

MENU

SEARCH

INDEX

DETAIL

INDEX

1 / 1

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-091704

(43)Date of publication of application : 28.03.2003

(51)Int.Cl.

G06K 19/07
B42D 15/10

(21)Application number : 2001-388727

(71)Applicant : HITACHI LTD

(22)Date of filing : 21.12.2001

(72)Inventor : MIZUSHIMA EIGA
TSUNODA MOTOYASU
HATANO TOMIHISA
KATAYAMA KUNIHIRO
TANAKA NORIO
TSUNEHIRO TAKASHI
KIMURA KOICHI

(30)Priority

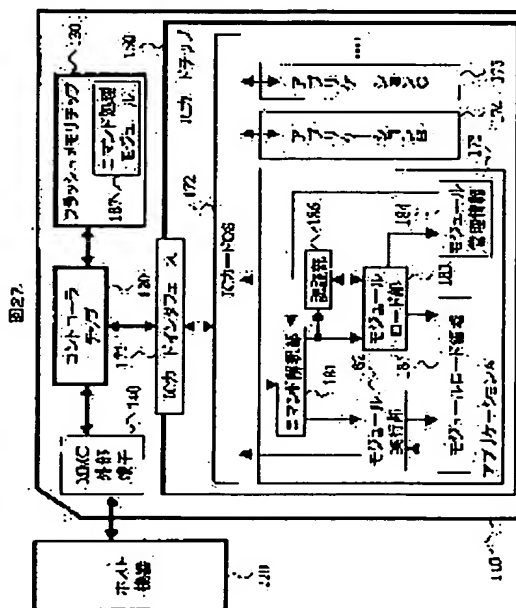
Priority number : 2001207212 Priority date : 09.07.2001 Priority country : JP

(54) STORAGE UNIT WITH NON-VOLATILE MEMORY AND INFORMATION PROCESSING DEVICE WITH FREELY DETACHABLE STORAGE UNIT

(57)Abstract:

PROBLEM TO BE SOLVED: To make an IC execute a large amount of processing even when the storage capacity in the IC (for example, ROM and EEROM) is small.

SOLUTION: This storage unit is provided with a flash memory chip 130, an IC card chip 150 capable of executing the security processing (encryption, decryption and the like), and a controller chip 12 for controlling the access to the flash memory chip and the IC card chip in accordance with the request from a host 220, a program of an IC card necessary for the security processing and a part of the data are stored in the flash memory chip, or the IC card chip requests the data processing utilizing the flash memory chip and the host equipment.



LEGAL STATUS

[Date of request for examination]

13.05.2004

2005 09 14 16:06

【特許請求の範囲】

【請求項1】 ホスト機器からのデータを記憶するためのデータ領域を有する不揮発性メモリと、セキュリティ処理を実行するためのICと、前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラとを備えた記憶装置において、

前記不揮発性メモリは、前記アプリケーションプログラムの一部を格納するための管理領域前記ICは、前記セキュリティ処理を前記ICに実行させるためのアプリケーションプログラムの他部を格納するための格納領域を有する記憶装置。

【請求項2】 前記ICは、前記アプリケーションプログラムの一部を暗号化し、前記コントローラは、暗号化された前記アプリケーションプログラムの一部を、前記不揮発性メモリに書き込む請求項1に記載の記憶装置。

【請求項3】 前記コントローラは、前記ホスト機器からのコマンドに応じて、前記不揮発性メモリから前記暗号化されたアプリケーションプログラムの一部を読み出し、前記ICへ転送し、前記ICは、前記暗号化されたアプリケーションプログラムの一部を受信し、復号化する請求項2に記載の記憶装置。

【請求項4】 前記ICは、中央処理装置と、前記ICのオフィスシステムプログラムを格納するための第1のROMと、前記格納領域を有する電氣的に書き換え可能な第2のROMと、前記アプリケーションプログラム及び前記オフィスシステムプログラムを格納するためのRAMとを有する請求項1に記載の記憶装置。

【請求項5】 前記セキュリティ処理は、暗号化又は復号化処理を含む請求項1に記載の記憶装置。

【請求項6】 データを記憶するためのデータ領域を有する不揮発性メモリとセキュリティ処理を実行するためのICと前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラとを有し且つ着脱自在な記憶装置と、前記セキュリティ処理を前記記憶装置へ要求するためのCPUと、ネットワークと通信するための通信装置とを備えた情報処理装置において、前記不揮発性メモリは、前記アプリケーションプログラムの一部を格納するための管理領域前記ICは、前記セキュリティ処理を前記ICに実行させるためのアプリケーションプログラムの他部を格納するための格納領域を有する情報処理装置。

【請求項7】 前記通信装置は、ネットワークを通じて、銀行取引を実行するための銀行取引サーバとクレジット決済を実行するためのクレジット決済サーバとコンテンツ配信を実行するためのコンテンツ配信サーバの少なくとも1つと通信可能である請求項6に記載の情報処理装置。

【請求項8】 データ領域を有する不揮発性メモリと、前

記不揮発性メモリに比較して耐タンパ性の高いICと、前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラとを備えた記憶装置において、前記ICは、銀行取引を処理するための第1のアプリケーションプログラムと、クレジット決済を処理するための第2のアプリケーションプログラムと、配信コンテンツを処理するための第3のアプリケーションプログラムとを有し、ホスト機器からのコマンドに応じて、前記第1のアプリケーションプログラム又は前記第2のアプリケーションプログラム又は前記第3のアプリケーションプログラムを選択し、実行する記憶装置。

【請求項9】 前記ICは、前記第1のアプリケーションプログラムを実行するための第1のICと、前記第2のアプリケーションプログラムを実行するための第2のICと、前記第3のアプリケーションプログラムを実行するための第3のICとを有する請求項8に記載の記憶装置。

【請求項10】 前記不揮発性メモリは、前記ホスト機器からのデータを格納するためのデータ領域と、前記第1のアプリケーションプログラムの一部と前記第2のアプリケーションプログラムの一部と前記第3のアプリケーションプログラムの一部との少なくとも1つを格納するための管理領域を有する請求項8に記載の記憶装置。

【請求項11】 不揮発性メモリと前記不揮発性メモリに比較して耐タンパ性の高いICと前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラとを有し且つ着脱自在な記憶装置と、前記記憶装置へコマンドを発行するためのCPUと、ネットワークと通信するための通信装置とを備えた情報処理装置において、前記ICは、銀行取引を処理するための第1のアプリケーションプログラムと、クレジット決済を処理するための第2のアプリケーションプログラムと、配信コンテンツを処理するための第3のアプリケーションプログラムとを有し、前記CPUからの前記コマンドに応じて、前記第1のアプリケーションプログラム又は前記第2のアプリケーションプログラム又は前記第3のアプリケーションプログラムを選択し、実行する情報処理装置。

【請求項12】 不揮発性メモリと、ICと、前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラと、前記不揮発性メモリと前記ICと前記コントローラとによって共有化され且つホスト機器と接続するためのインタフェースを備えた記憶装置において、前記コントローラは、前記ホスト機器からの第1のコマンドを受信し、前記ホスト機器からの第1のコマンドに応じて、前記ICが解釈可能な第2のコマンドを作成し、前記ICへ送信する記憶装置。

【請求項13】 不揮発性メモリと、処理の実行に必要なワークメモリを有するICと、前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラとを備えた記憶装置において、

前記コントローラは、前記ICが解釈可能なコマンドを発行し、

前記ICは、前記コマンドによって処理対象とされたデータのデータ量と前記ワームメモリの空き容量とを比較し、その比較結果に応じて前記コントローラへのレスポンスを決定し、前記コントローラへ送信する記憶装置。

【請求項14】前記ICは、前記データ量が前記空き容量より大きい場合に、前記コントローラへの処理要求を含む前記レスポンスを決定する請求項13に記載の記憶装置。

【請求項15】前記レスポンスは、前記ICと前記コントローラとの間で予め定義されたステータスワードを含む請求項14に記載の記憶装置。

【請求項16】前記コントローラは、前記レスポンスを受信し、前記レスポンスによって要求された処理を実行し、その処理結果を前記ICへ送信する請求項14に記載の記憶装置。

【請求項17】前記ICは、前記データ量が前記空き容量以下の場合に、前記データに対し前記コマンドによって要求された処理を実行し、その処理結果を前記コントローラへ送信する請求項14に記載の記憶装置。

【請求項18】不揮発性メモリとICと前記不揮発性メモリ及び前記ICへのアクセスを制御するためのコントローラとを有し且つ着脱自在な記憶装置と、CPUと、ネットワークと通信するための通信装置とを備えた情報処理装置において、

前記コントローラは、前記ICが解釈可能なコマンドを発行し、

前記ICは、前記コマンドによって処理対象とされたデータのデータ量と前記ワームメモリの空き容量とを比較し、その比較結果に応じて前記コントローラへのレスポンスを決定し、前記コントローラへ送信する情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はセキュリティ機能を搭載した記憶装置及びその記憶装置が挿入可能なHOST機器及びその記憶装置が挿入されたHOST機器に係り、不揮発性のフラッシュメモリチップ、IC(Integrated circuit;集積回路)カードチップ及びコントローラチップとを有するメモリカード及びそのメモリカードが挿入可能な(着脱自在な)情報処理装置及びそのメモリカードが挿入された情報処理装置に関する。

【0002】

【従来の技術】ICカードは、プラスチックカード基板中にIC(集積回路)チップを埋め込んだものであり、その表面にICチップの外部端子を持つ。ICチップの外部端子には電源端子、クロック端子、データ入出力端子などがある。ICチップは、接続装置が外部端子から電源や駆動クロックを直接供給することによって動作す

る。ICカードは外部端子を通して端末機などの接続装置との間で電気信号を送受信することにより、接続装置と情報交換をおこなう。情報交換の結果として、ICカードは計算結果や記憶情報の送出、記憶情報の変更をおこなう。ICカードは、これらの動作仕様に基づいて、機密データ保護や個人認証などのセキュリティ処理を実行する機能を持つことができる。ICカードは、クレジット決済やバンキングなど機密情報のセキュリティが必要とされるシステムにおいて、個人識別のためのユーザデバイスとして利用されている。

【0003】

【発明が解決しようとする課題】セキュリティシステムにおいて利用されるユーザ識別装置は、秘密情報を用いて演算を行う際に、その秘密情報あるいはその秘密情報を推定できるような情報を外にももらさないように設計される必要がある。すなわち、耐タンパ性を持つことが必要とされる。また、ユーザ識別装置はユーザにとって利便性が高いことが必要である。例えば、1台の装置でより多くのセキュリティシステムに対応できるしくみを持つこと、さらに、より大きなサイズのデータを処理する能力を持つことである。

【0004】本発明の目的は、ICが実行するためのプログラムやデータ、ICを管理するための情報をIC外部の不揮発性メモリに保持するため、IC内の記憶容量が小さい場合にも、ICが多くの処理を実行できる記憶装置及び情報処理装置を提供することである。

【0005】本発明の目的は、ICが実行する処理の一部をIC外部のコントローラが実行するため、IC内の記憶容量(例えば、RAM)が小さい場合にも、ICが多くの処理を実行できる記憶装置及び情報処理装置を提供することである。

【0006】

【課題を解決するための手段】本発明は、不揮発性メモリが、ICに実行させるためのアプリケーションプログラムの一部を格納する。

【0007】本発明は、ICが、銀行取引を処理するための第1のアプリケーションプログラムと、クレジット決済を処理するための第2のアプリケーションプログラムと、配信コンテンツを処理するための第3のアプリケーションプログラムとを有し、HOST機器からのコマンドに応じて、第1のアプリケーションプログラム又は第2のアプリケーションプログラム又は第3のアプリケーションプログラムを選択し、実行する。

【0008】本発明は、コントローラが、HOST機器からの第1のコマンドに応じて、ICが解釈可能な第2のコマンドを作成し、ICへ送信する。

【0009】本発明は、ICが、コントローラからのコマンドによって処理対象とされたデータのデータ量とIC内のワームメモリの空き容量とを比較し、その比較結果に応じてコントローラへのレスポンスを決定し、コン

トローラへ送信する。

【0010】

【発明の実施の形態】以下、本発明の一実施形態について説明する。

【0011】図22は、本発明を適用したMultiMediaCard (MultiMediaCardはInfineon Technologies AGの登録商標である。以下、「MMC」と略記する。)の内部構成図を簡単に表したものである。MMC110は、MultiMediaCard仕様に準拠するのが好ましい。MMC110は、外部に接続したホスト機器220がMultiMediaCard仕様に準拠したメモリカードコマンドを発行することによって、機密データ保護や個人認証などに必要な暗号演算をおこなうセキュリティ処理機能を持つ。ホスト機器220は、例えば、携帯電話、携帯情報端末(PDA)、パーソナルコンピュータ、音楽再生(及び録音)装置、カメラ、ビデオカメラ、自動預金預払器、街角端末、決済端末等が該当する。MMC110は、MMC外部端子140、コントローラチップ120、フラッシュメモリチップ130、ICカードチップ150を持つ。フラッシュメモリチップ130は、不揮発性の半導体メモリを記憶媒体とするメモリチップであり、フラッシュメモリコマンドによりデータの読み書きができる。MMC外部端子140は7つの端子から構成され、外部のホスト機器220と情報交換するために、電源供給端子、クロック入力端子、コマンド入出力端子、データ入出力端子、グランド端子を含む。コントローラチップ120は、MMC110内部の他の構成要素(MMC外部端子140、フラッシュメモリチップ130、ICカードチップ150)と接続されており、これらを制御するマイコンチップである。ICカードチップ150は、ICカードのプラスチック基板中に埋め込むためのマイコンチップであり、その外部端子、電気信号プロトコル、コマンドはISO/IEC7816規格に準拠している。ICカードチップ150の外部端子には、電源供給端子、クロック入力端子、リセット入力端子、I/O入出力端子、グランド端子がある。コントローラチップ120は、ICカードチップ150の外部端子からICカードチップ150にICカードコマンドを発行することによって、外部のホスト機器220から要求されたセキュリティ処理に必要な演算をおこなう。

【0012】図26は、本発明のICカードチップの内部構成を示す図である。ICカードチップ150は、演算処理を行うためのCPU(マイコン)158と、データ(プログラムを含む。)を記憶するためのROM(Read Only Memory)159とRAM(Random Access Memory)160とEEPROM(Electrically Erasable Programmable ROM)162と、

暗号/復号に関する処理を行うための暗号コプロセッサ163と、外部とデータを送受信するためのシリアルインターフェース161とを備え、それらは、バス164によって接続される。そして、その暗号コプロセッサ163によって、ホスト機器220からのコマンドに応じて、ICカードチップ150自らが、セキュリティ処理を実行することが可能である。尚、暗号コプロセッサ163(ハードウェア)の替わりに、プログラム(ソフトウェア)に従ってCPU158がセキュリティ処理を実行してもよい。

【0013】一方、フラッシュメモリチップ130には、記憶素子を備えるが、マイコンは存在しない。

【0014】セキュリティ処理は、例えば、ICカードチップ150内の記憶領域にデータが書き込まれるとき、又は、ICカードチップ150内の記憶領域からデータが読み出されるときに実行される。ICカードチップ150のEEPROMの記憶容量は、フラッシュメモリチップ130の記憶容量より小さい。但し、ICカードチップ150のEEPROMの記憶容量は、フラッシュメモリチップ130の記憶容量と同じでもよいし、大きくてもよい。

【0015】ICカードチップ150には、セキュリティ評価基準の国際標準であるISO/IEC15408の評価・認証機関によって認証済みである製品を利用する。一般に、セキュリティ処理をおこなう機能を持つICカードを実際の電子決済サービスなどで利用する場合、そのICカードはISO/IEC15408の評価・認証機関による評価と認定を受ける必要がある。MMCにセキュリティ処理をおこなう機能を追加することによってMMC110を実現し、それを実際の電子決済サービスなどで利用する場合、MMC110も同様にISO/IEC15408の評価・認証機関による評価と認定を受ける必要がある。本発明によれば、MMC110は、評価・認証機関によって認証済みのICカードチップ150を内蔵し、そのICカードチップ150を利用してセキュリティ処理をおこなう構造を持つことにより、セキュリティ処理機能を得る。したがって、MMC110はISO/IEC15408に基づくセキュリティ評価基準を容易に満足することができ、MMCにセキュリティ処理機能を追加するための開発期間を短縮することができる。

【0016】MMC110は、MultiMediaCard仕様に準拠した外部インタフェースを持つのが好ましい。MMC110は、一種類の外部インタフェースを通じて、標準メモリカードコマンド(フラッシュメモリチップ130へアクセスするためのコマンド)に加えて、セキュリティ処理を実行するコマンドを受け付ける必要がある。コントローラチップ120は、MMC110が受信したコマンドが標準メモリカードコマンドであるか、セキュリティ処理を実行するコマンドであるかに

よって、アクセスすべきチップを選択し、コマンド処理を分配する機能を持つ。本発明によれば、標準メモリカードコマンドを受信したならば、フラッシュメモリチップ130を選択し、これにフラッシュメモリコマンドを発行してホストデータを読み書きできる。また、セキュリティ処理を実行するコマンドを受信したならば、ICカードチップ150を選択し、これにICカードコマンドを発行してセキュリティ処理を実行することができる。

【0017】ICカードチップ150の外部端子は、グランド端子を除いて、電源供給端子、クロック入力端子、リセット入力端子、I/O入出力端子がコントローラチップ120に接続されている。

【0018】コントローラチップ120は、電源供給端子、クロック入力端子を通して、ICカードチップ150への電源供給、クロック供給を制御する。本発明によれば、ホスト機器220からセキュリティ処理を要求されないときには、ICカードチップ150への電源供給やクロック供給を停止させることができ、MMC110の電力消費を削減することができる。

【0019】電源供給されていないICカードチップ150を、ICカードコマンドを受信できる状態にするには、まず、ICカードチップ150に電源供給を開始し、リセット処理を施すことが必要である。コントローラチップ120は、MMC110がホスト機器220からセキュリティ処理を実行するコマンドを受信したのを契機に、電源供給端子を通してICカードチップ150への電源供給を開始する機能を持つ。また、コントローラチップ120は、MMC110がホスト機器220からセキュリティ処理を実行するコマンドを受信したのを契機に、リセット入力端子を通してICカードチップ150のリセット処理をおこなう機能を持つ。本発明によれば、コントローラチップ120は、セキュリティ処理を実行するコマンドを受信するまでICカードチップ150への電源供給を停止させておくことができる。したがって、MMC110の電力消費を削減することができる。

【0020】コントローラチップ120は、ICカードチップ150のクロック入力端子を通してICカードチップ150に供給するクロック信号をMMC110内部で発生し、その周波数、供給開始タイミング、供給停止タイミングを制御する機能を持つ。本発明によれば、MMC外部端子140のクロック入力端子のクロック信号と無関係にすることができるため、ホスト機器220によるタイミング解析、電力差分析、故障利用解析と呼ばれる攻撃法に対してセキュリティが向上する。

【0021】図21は、フラッシュメモリチップ130の詳細な内部構成を表している。フラッシュメモリチップ130は、ホストデータ領域2115と管理領域2110を含む。ホストデータ領域2115は、セクタ単

位に論理アドレスがマッピングされている領域であり、ホスト機器220が論理アドレスを指定してデータを読み書きできる領域である。ホストデータ領域2115は、ユーザファイル領域2130とセキュリティ処理アプリケーション領域2120を含む。ユーザファイル領域2130は、ユーザが自由にファイルデータを読み書きできる領域である。セキュリティ処理アプリケーション領域2120は、ホスト機器220がセキュリティ処理アプリケーションに必要なデータを格納する領域であり、ユーザが不正にアクセスしないように、ホスト機器220のセキュリティ処理アプリケーションが論理的にユーザアクセス制限をかける。ここに格納するデータとしては、ホスト機器220のアプリケーションプログラム、そのアプリケーション専用のデータ、セキュリティ処理に使用される証明書など（例えば、電子決済アプリケーションプログラム、電子決済ログ情報、電子決済サービス証明書など）が可能である。本発明によれば、MMC110が、ホスト機器220がセキュリティ処理をおこなう上で使用するデータをホスト機器220の代わりに格納するため、ホスト機器220にとって利便性が向上する。

【0022】一方、管理領域2110は、コントローラチップ120がICカードチップ150を管理するための情報を格納する領域である。管理領域2110は、ICカード制御パラメータ領域2111、ICカード環境設定情報領域2112、CLK2設定情報領域2113、セキュリティ処理バッファ領域2114、セキュリティ処理ステータス領域2116とを含む。2111～2116の領域の詳細な使用方法については後述する。

【0023】コントローラチップ120は、フラッシュメモリチップ130の管理領域2110のセキュリティ処理バッファ領域2114を、ICカードチップ150でセキュリティ処理を実行する際のメインメモリまたはバッファメモリとして利用する。ホスト機器220がセキュリティ処理を実行するコマンドによりMMC110にアクセスした際に、MMC110がホスト機器220からICカードチップ150に一度に送信できないほどの大きなサイズのセキュリティ関連データを受信したならば、コントローラチップ120はフラッシュメモリチップ130へのアクセスを選択し、そのデータを十分な容量を持つセキュリティ処理バッファ領域2114に一時的に格納する。ICカードチップ150に一度に送信できないほどのサイズは、ICカードコマンドの許容データサイズ（例えば、255バイト又は256バイト）を超えるサイズである。そして、コントローラチップ120はそれをICカードチップ150に送信できるサイズのデータに分割し、分割データをフラッシュメモリチップ130から読み出し、段階的にICカードチップ150に送信する。つまり、分割されたデータの読み出し、書き込みを繰り返す。本発明によれば、ホスト機器

220にとって、大きなサイズのセキュリティ関連データを扱うことができるので、セキュリティ処理の利便性が向上する。

【0024】上記のセキュリティ処理バッファ領域2114を含む管理領域2110は、ホスト機器220が不正にアクセスしてセキュリティ処理を解析することができないように、コントローラチップ120により物理的にホストアクセス制限がかけられている。つまり、管理領域2110はホスト機器220が直接データを読み書きできない。本発明によれば、ホスト機器220がセキュリティ処理バッファ領域2114の内容を自由に読み出ししたり改ざんすることができないため、セキュリティ処理の信頼性や安全性が向上する。

【0025】図23は、MMC110を利用したセキュリティ処理の一例として、コンテンツ配信のセキュリティ処理を表したものである。コンテンツプロバイダ2310は、MMC110を所有するユーザにコンテンツ2314を販売する業者である。ホスト機器220は、この例では、コンテンツプロバイダ2310とネットワークなどを介して接続することができる端末機である。ユーザはMMC110をホスト機器220に接続してコンテンツ2314を購入する。以下、その手順を説明する。まず、ホスト機器220はMMC110に、フラッシュメモリチップ130に格納されたユーザ証明書2321を読み出すコマンドを発行する。MMC110のコントローラチップ120は、フラッシュメモリチップ130のセキュリティ処理アプリケーション領域2120に格納されたユーザ証明書2321を読み出し、それをホスト機器220に送信する。そして、ホスト機器220はそれをコンテンツプロバイダ2310に送信する。コンテンツプロバイダ2310はユーザ証明書2321につけられたデジタル署名を検証する(2311)。検証が成功したならば、乱数発生器によりセッション鍵を生成し(2312)、それをユーザ証明書2321から抽出したユーザ公開鍵によって暗号化する(2313)。さらに、コンテンツ2314をそのセッション鍵によって暗号化する(2315)。コンテンツプロバイダ2310はステップ2313の結果をホスト機器220に送信する。ホスト機器220は、ステップ2313の結果をユーザ秘密鍵2322によって復号するセキュリティ処理を要求するコマンドを、MMC110に発行する。コントローラチップ120は、ステップ2313の結果をユーザ秘密鍵2322によって復号するICカードコマンドを、ICカードチップ150に発行する。ICカードチップ150は、ユーザ秘密鍵2322によってステップ2313の結果を復号して、セッション鍵を取得する(2323)。ホスト機器220は、この復号処理が成功したかを示す情報を出力させるコマンドをMMC110に発行する。コントローラチップ120は、ICカードチップ150の出力する復号結果(復号

処理が成功したかを示すICカードレスポンス)をもとにしてホスト機器220の求める情報を構築する。そして、MMC110はその情報をホスト機器220に送信する。次に、コンテンツプロバイダ2310は、ステップ2315の結果を、ホスト機器220に送信する。ホスト機器220は、ステップ2313の結果をセッション鍵(ステップ2323によって取得した鍵)によって復号するセキュリティ処理を要求するコマンドを、MMC110に発行する。コントローラチップ120は、ステップ2315の結果をセッション鍵によって復号するICカードコマンドを、ICカードチップ150に発行する。ICカードチップ150は、セッション鍵によってステップ2315の結果を復号して、コンテンツ2314を復元する(2324)。コントローラチップ120は、このコンテンツ2314をICカードチップ150から受信し、フラッシュメモリチップ130に書き込む。ホスト機器220は、この復号処理が成功したかを示す情報を出力させるコマンドをMMC110に発行する。コントローラチップ120は、ICカードチップ150の出力する復号結果(復号処理が成功したかを示すICカードレスポンス)をもとにしてホスト機器220の求める情報を構築する。そして、MMC110はその情報をホスト機器220に送信する。ホスト機器220が、コンテンツを無事に受信したことをコンテンツプロバイダ2310に伝え、コンテンツプロバイダ2310はユーザ証明書に記載されたユーザにコンテンツ料金を課金する。ユーザは、ホスト機器220でMMC110内のフラッシュメモリチップ130に格納されたコンテンツ2314を読み出して利用することができる。また、フラッシュメモリチップ130の記憶媒体に大容量のフラッシュメモリを使用すれば、多くのコンテンツを購入できる。本発明によれば、コンテンツ配信におけるセキュリティ処理とコンテンツ蓄積の両方をMMC110によって容易に実現できる。コンテンツ料金の決済を、ICカードチップ150を利用して行ってもよい。

【0026】図24と図25は、それぞれ、本発明をSDカード(幅24ミリメートル、長さ32ミリメートル、厚さ2.1ミリメートルで、9つの外部端子を持ち、フラッシュメモリを搭載した小型メモリカードである。)とメモリスティック(メモリスティックはソニー株式会社の登録商標である。)に適用したときの簡単な内部構成図を表したものである。本発明を適用したSDカード2410は、SDカードコントローラチップ2420、フラッシュメモリチップ2430、SDカード外部端子2440、ICカードチップ150を含む。本発明を適用したメモリスティック2510は、メモリスティックコントローラチップ2520、フラッシュメモリチップ2530、メモリスティック外部端子2540、ICカードチップ150を含む。フラッシュメモリチップ2430と2530は、不揮発性の

半導体メモリを記憶媒体とするメモリチップであり、フラッシュメモリコマンドによりデータの読み書きができる。SDカードコントローラチップ2420とメモリースティックコントローラチップ2520はそれぞれSDカードとメモリースティック内の他の構成要素を制御するマイコンチップである。

【0027】SDカード外部端子2440は9つの端子からなり、それらの位置は、端からData2端子2441、Data3端子2442、Com端子2443、Vss端子2444、Vdd端子2445、Clock端子2446、Vss端子2447、Data0端子2448、Data1端子2449の順で並んでいる。Vdd端子2445は電源供給端子、Vss端子2444と2447はグラウンド端子、Data0端子2448とData1端子2449とData2端子2441とData3端子2442はデータ入出力端子、Com端子2443はコマンド入出力端子、Clock端子2446はクロック入力端子である。SDカード2410は、外部に接続するSDカードホスト機器2460とのインタフェース仕様にMMC110と違いがあるものの、MMC外部端子140と非常に類似した外部端子を持ち、MMC110と同様に外部からコマンドを発行することにより動作する特徴を持つため、本発明を適用することができる。

【0028】一方、メモリースティック外部端子2540は10個の端子からなり、それらの位置は、端からGnd端子2541、BS端子2542、Vcc端子2543、予約端子Rsvを1つ飛ばしてDIO端子2544、INS端子2545、予約端子Rsvを1つ飛ばしてSCK端子2546、Vcc端子2547、Gnd端子2548の順で並んでいる。Vcc端子2543と2547は電源供給端子、Gnd端子2541と2548はグラウンド端子、DIO端子2544はコマンドおよびデータ入出力端子、SCK端子2546はクロック入力端子である。メモリースティック2510は、外部に接続するメモリースティックホスト機器2560とのインタフェース仕様にMMC110と違いがあるものの、MMC110と同様に外部からコマンドを発行することにより動作する特徴を持つため、本発明を適用することができる。

【0029】図1は、本発明を適用したMMCの詳細な内部構成図を表したものである。また、図2は、図1のMMC110と接続したホスト機器220の構成とその接続状態を表したものである。ホスト機器220は、VCC1電源221、CLK1発振器222、ホストインタフェース223を持つ。

【0030】MMC110は、外部のホスト機器220と情報交換するためのMMC外部端子140を持つ。MMC外部端子140は、CS端子141、CMD端子142、GND1端子143および146、VCC1端子

144、CLK1端子145、DAT端子147の7つの端子とを含む。MultiMediaCard仕様は、MMCの動作モードとしてMMCモードとSPIモードという2種類を規定しており、動作モードによってMMC外部端子140の使用法は異なる。本実施例ではMMCモードでの動作の場合について詳細に説明する。VCC1端子144は、VCC1電源221と接続されており、ホスト機器220がMMC110に電力を供給するための電源端子である。GND1端子143および146は、VCC1電源221と接続されており、MMC110の電氣的なグラウンド端子である。GND1端子143とGND1端子146は、MMC110内部で電氣的に短絡されている。CS端子141は、ホストインタフェース223に接続されており、SPIモードの動作において使用される入力端子である。ホスト機器220が、MMC110にSPIモードでアクセスするときには、CS端子141にLレベルを入力する。MMCモードの動作では、CS端子141を使用する必要はない。CMD端子142は、ホストインタフェース223に接続されており、ホスト機器220が、メモリカードインタフェース仕様に準拠したメモリカードコマンドをMMC110に送信したり、同仕様に準拠したメモリカードレスポンスをMMC110から受信するために使用する入出力端子である。DAT端子147は、ホストインタフェース223に接続されており、ホスト機器220が、メモリカードインタフェース仕様に準拠した形式の入力データをMMC110に送信したり、同仕様に準拠した形式の出力データをMMC110から受信するために使用する入出力端子である。CLK1端子145は、CLK1発振器222に接続されており、CLK1発振器222が生成するクロック信号が入力される端子である。ホスト機器220が、CMD端子142を通してメモリカードコマンド、メモリカードレスポンスを受信したり、DAT端子147を通してホストデータを送受信するときに、CLK1端子145にクロック信号が入力される。ホストインタフェース223には、CLK1発振器222からクロック信号が供給されており、メモリカードコマンド、メモリカードレスポンス、ホストデータは、CLK1発振器222が生成するクロック信号にビット単位で同期して、ホスト機器220とMMC110との間を転送される。

【0031】MMC110は、コントローラチップ120を持つ。コントローラチップ120は、CPU121、フラッシュメモリI/F制御回路122、MMC I/F制御回路123、CLK0発振器124、VCC2生成器125、VCC2制御回路126、CLK2制御回路127、ICカードI/F制御回路128とを含む。これらの構成要素121~128は、ホスト機器220からVCC1端子144やGND1端子143、146を通して供給された電力により動作する。MMC I

／F制御回路123は、CS端子141、CMD端子142、CLK1端子145、DAT端子147と接続されており、MMC110がそれらの端子を通してホスト機器220と情報交換するためのインタフェースを制御する論理回路である。CPU121は、MMC I／F制御回路123と接続されており、MMC I／F制御回路123を制御する。MMC I／F制御回路123がCMD端子142を通してホスト機器220からメモリカードコマンドを受信すると、MMC I／F制御回路123はそのコマンドの受信が成功したかどうかの結果をホスト機器220に伝えるためCMD端子142を通してホスト機器220にレスポンスを送信する。CPU121は、受信したメモリカードコマンドを解釈し、コマンド内容に応じた処理を実行する。また、そのコマンド内容に応じてホスト機器220とDAT端子147を通してデータの送受信をおこなう必要がある場合、CPU121は、MMC I／F制御回路123へのデータの送出、MMC I／F制御回路123からのデータの取得をおこなう。さらに、CPU121は、MMC I／F制御回路123とホスト機器220との間のデータ転送手続きも制御する。例えば、ホスト機器220から受信したデータの処理中に、ホスト機器220がMMC110への電源供給を停止することがないように、CPU121はDAT端子147にLレベルを出力させ、MMC110がビジー状態であることをホスト機器220に伝える。CLK0発振器124は、CPU121と接続され、CPU121を動作させる駆動クロックを供給する。

【0032】MMC110は、フラッシュメモリチップ130を持つ。フラッシュメモリチップ130は、不揮発性の半導体メモリを記憶媒体とするメモリチップである。フラッシュメモリチップ130は、ホスト機器220からVCC1端子144やGND1端子143、146を通して供給された電力により動作する。フラッシュメモリチップ130は、外部からのフラッシュメモリコマンドに従って、入力されたデータを不揮発性の半導体メモリに格納するライト機能、また同メモリに格納されたデータを外部に出力するリード機能を持つ。フラッシュメモリ I／F制御回路122は、フラッシュメモリチップ130にフラッシュメモリコマンドを発行したり、そのコマンドで入出力するデータを転送するための論理回路である。CPU121は、フラッシュメモリ I／F制御回路122を制御し、フラッシュメモリチップ130にデータのライト機能やリード機能を実行させる。ホスト機器220から受信したデータをフラッシュメモリチップ130にライトしたり、フラッシュメモリチップ130に格納されたデータをホスト機器220に送信する必要があるとき、CPU121は、フラッシュメモリ I／F制御回路122とMMC I／F制御回路123の間のデータ転送を制御する。

【0033】MMC110は、ICカードチップ150

を持つ。ICカードチップ150は、ICカードの基板中に埋め込むことを目的として設計されたマイコンチップであり、ICカードの外部端子規格に準拠した8つの外部端子を持つ。このうち6つの端子は、ICカードの外部端子規格により使用法が割り付けられており、残りの2つは将来のための予備端子である。その6つの端子は、VCC2端子151、RST端子152、CLK2端子153、GND2端子155、VPP端子156、I／O端子157である。

【0034】ICカードチップ150のグランド端子は、MMC外部端子140のGND1（グランド端子）146に接続される。ICカードチップ150のVCC2端子（電源入力端子）151は、コントローラチップ120のVCC2制御回路126に接続される。ICカードチップ150のRST端子（リセット入力端子）152とI／O端子（データ入出力端子）157は、コントローラチップ120のICカード I／F制御回路128に接続される。ICカードチップ150のCLK2端子（クロック入力端子）153は、コントローラチップ120のCLK2制御回路127に接続される。

【0035】フラッシュメモリチップ130のVCC端子（電源入力端子）は、MMC外部端子140のVCC1144に接続される。フラッシュメモリチップ130のVSS端子（グランド端子）は、MMC外部端子140のGND1146に接続される。フラッシュメモリチップ130のI／O端子（データ入出力端子）とレディ／ビジー端子とチップイネーブル端子とアウトプットイネーブル端子とライトイネーブル端子とクロック端子とリセット端子とは、コントローラチップ120のフラッシュメモリ I／F制御回路122に接続される。

【0036】VCC2端子151は、ICカードチップ150に電力を供給するための電源端子である。VCC2制御回路126は、MOS-FET素子を用いたスイッチ回路によりVCC2端子151への電力の供給開始と供給停止を制御する回路である。VCC2生成器125はVCC2端子151に供給する電圧を発生し、それをVCC2制御回路126に供給する。ICカードの電気信号規格はICカードの動作クラスとしてクラスAとクラスBを規定している。VCC2端子151に供給する標準電圧は、クラスAでは5V、クラスBでは3Vである。本発明はICカードチップ150の動作クラスによらず適用できるが、本実施例ではICカードチップ150がクラスBで動作する場合について詳細に説明する。VPP端子156は、ICカードチップ150がクラスAで動作する時に、内部の不揮発性メモリにデータを書き込んだり消去したりするために使用される可変電圧を供給する端子であり、クラスBで動作する時には使用しない。GND2端子155は、ICカードチップ150の電氣的なグランド端子であり、GND1端子143、146と短絡されている。VCC2制御回路126

はCPU121と接続され、CPU121はVCC2端子151への電力供給の開始と停止を制御することができる。ICカードチップ150を使用しないときは、CPU121はVCC2端子151への電力供給を停止することができる。MMC110は、ICカードチップ150への電力供給を停止することにより、それが消費する電力を節約することができる。ただし、電力供給を停止すると、ICカードチップ150の内部状態は、ICカードチップ150内部の不揮発性メモリに記憶されたデータを除いて維持されない。

【0037】CLK2端子153は、ICカードチップ150にクロック信号を入力する端子である。

【0038】CLK2制御回路127は、CLK2端子153にクロックを供給する回路である。CLK2制御回路127は、CLK0発振器124から供給されたクロック信号をもとにしてCLK2端子153に供給するクロック信号を生成する。CLK2制御回路127はCPU121と接続されており、CLK2端子153へのクロックの供給開始と供給停止をCPU121から制御することができる。ICカードチップ150は、自身内部に駆動クロック発振器をもたない。そのため、CLK2端子153から駆動クロックを供給することによって動作する。CLK2制御回路127が、CLK2端子153へのクロック供給を停止すると、ICカードチップ150の動作は停止するため、ICカードチップ150の消費電力を低下させることができる。この時、VCC2端子151への電力供給が保たれていれば、ICカードチップ150の内部状態は維持される。ここで、CLK2端子153に供給するクロック信号の周波数をF2、CLK0発振器124から供給されたクロック信号の周波数をF0、PとQを正の整数とすると、CLK2制御回路127は、 $F2 = (P/Q) * F0$ の関係になるようなクロック信号を作成して、これをCLK2端子153に供給する。PとQの値はCPU121により設定できるようになっている。Pを大きく設定してF2を大きくすると、ICカードチップ150の内部処理をより高速に駆動できる。Qを大きく設定してF2を小さくすると、ICカードチップ150の内部処理はより低速に駆動され、ICカードチップ150の消費電力を低下させることができる。ICカードチップ150の駆動クロック周波数は、ICカードチップ150が正しく動作できるように許容周波数範囲内に設定される必要がある。そのため、CLK2制御回路127は、F2の値がその許容周波数範囲を外れるようなPとQの値を設定させない特徴を持つ。

【0039】I/O端子157は、ICカードチップ150にICカードコマンドを入力したり、ICカードチップ150がICカードレスポンスを出力するときに使用する入出力端子である。ICカードI/F制御回路128は、I/O端子157と接続されており、I/O端

子157を通してICカードコマンドの信号送信やICカードレスポンスの信号受信をおこなう回路である。ICカードI/F制御回路128はCPU121に接続されており、CPU121は、ICカードI/F制御回路128によるICカードコマンドやICカードレスポンスの送受信の手続きを制御したり、送信すべきICカードコマンドデータをICカードI/F制御回路128に設定したり、受信したICカードレスポンスをICカードI/F制御回路128から取得する。ICカードI/F制御回路128にはCLK2制御回路127からクロックが供給されており、ICカードコマンドやICカードレスポンスは、CLK2端子153に供給するクロック信号にビット単位で同期して、I/O端子157を通して送受信される。また、RST端子152は、ICカードチップ150をリセットするときリセット信号を入力する端子である。ICカードI/F制御回路128は、RST端子152と接続されており、CPU121の指示によりICカードチップ150にリセット信号を送ることができる。

【0040】ICカードチップ150は、ICカードの電気信号規格やコマンド規格に基づいて情報交換をおこなう。ICカードチップ150へのアクセスパターンは4種類であり、図3～図6を用いて各パターンを説明する。図3は、CPU121の指示によりICカードチップ150が非活性状態（電源が遮断されている状態）から起動して内部状態を初期化するプロセス（以下、コールドリセットと呼ぶ）において、ICカードチップ150の外部端子の信号波形をシンプルに表したものである。図4は、CPU121の指示によりICカードチップ150が活性状態（電源が供給されている状態）で内部状態を初期化するプロセス（以下、ウォームリセットと呼ぶ）において、ICカードチップ150の外部端子の信号波形をシンプルに表したものである。図5は、CPU121の指示によりICカードチップ150にICカードコマンドを送信しICカードチップ150からICカードレスポンスを受信するプロセスにおいて、ICカードチップ150の外部端子の信号波形をシンプルに表したものである。図6は、CPU121の指示によりICカードチップ150を非活性状態にするプロセスにおいて、ICカードチップ150の外部端子の信号波形をシンプルに表したものである。図3～図6において、時間の方向は左から右にとっており、上の行から下の行に向かってVCC2端子151、RST端子152、CLK2端子153、I/O端子157で観測される信号を表す。また、破線はそれぞれの信号の基準（Lレベル）を表す。

【0041】図3を参照して、ICカードチップ150のコールドリセット操作を説明する。まず、ICカードI/F制御回路128はRST端子152をLレベルにする（301）。次に、VCC2制御回路126はVC

C2端子への電源供給を開始する(302)。次に、CLK2制御回路127はCLK2端子153へのクロック信号の供給を開始する(303)。次に、ICカードI/F制御回路128はI/O端子157を状態Z(プルアップされた状態)にする(304)。次に、ICカードI/F制御回路128はRST端子152をHレベルにする(305)。次に、ICカードI/F制御回路128はI/O端子157から出力されるリセット応答の受信を開始する(306)。リセット応答の受信が終了したら、CLK2制御回路127はCLK2端子153へのクロック信号の供給を停止する(307)。これで、コールドリセットの操作が完了する。なお、ステップ307は消費電力を低下させるための工夫であり、省略してもよい。

【0042】図4を参照して、ICカードチップ150のウォームリセット操作を説明する。まず、CLK2制御回路127はCLK2端子153へのクロック信号の供給を開始する(401)。次に、ICカードI/F制御回路128はRST端子152をLレベルにする(402)。次に、ICカードI/F制御回路128はI/O端子157を状態Zにする(403)。次に、ICカードI/F制御回路128はRST端子152をHレベルにする(404)。次に、ICカードI/F制御回路128はI/O端子157から出力されるリセット応答の受信を開始する(405)。リセット応答の受信が終了したら、CLK2制御回路127はCLK2端子153へのクロック信号の供給を停止する(406)。これで、ウォームリセットの操作が完了する。なお、ステップ406は消費電力を低下させるための工夫であり、省略してもよい。

【0043】図5を参照して、ICカードチップ150にICカードコマンドを送信しICカードチップ150からICカードレスポンスを受信する操作を説明する。まず、CLK2制御回路127はCLK2端子153へのクロック信号の供給を開始する(501)。なお、クロックがすでに供給されている場合、ステップ501は不要である。次に、ICカードI/F制御回路128はI/O端子157にコマンドデータの送信を開始する(502)。コマンドデータの送信が終了したら、ICカードI/F制御回路128はI/O端子157を状態Zにする(503)。次に、ICカードI/F制御回路128はI/O端子157から出力されるレスポンスデータの受信を開始する(504)。レスポンスデータの受信が終了したら、CLK2制御回路127はCLK2端子153へのクロック信号の供給を停止する(505)。これで、ICカードコマンド送信とICカードレスポンス受信の操作が完了する。なお、ステップ505は、消費電力を低下させるための工夫であり、省略してもよい。

【0044】図6を参照して、ICカードチップ150

を非活性化する操作を説明する。まず、CLK2制御回路127はCLK2端子153をLレベルにする(601)。次に、ICカードI/F制御回路128はRST端子152をLレベルにする(602)。次に、ICカードI/F制御回路128はI/O端子157をLレベルにする(603)。最後に、VCC2制御回路126はVCC2端子への電源供給を停止する(604)。これで、非活性化の操作が完了する。

【0045】ICカードチップ150は、機密データ保護や個人認証などに必要な暗号演算をおこなうセキュリティ処理機能を持つ。ICカードチップ150は、CPU121との間でICカードコマンドやICカードレスポンスの送受信することにより情報交換をおこない、その結果として、計算の結果や記憶されている情報の送
出、記憶されている情報の変更などをおこなう。CPU121は、ICカードチップ150を利用してセキュリティ処理を実行することができる。MMC110がホスト機器220から特定のメモリカードコマンドを受信すると、CPU121はそれを契機として、VCC2制御回路126を通してICカードチップ150への電源供給を制御したり、またはCLK2制御回路127を通してICカードチップ150へのクロック供給を制御したり、またはICカードI/F制御回路128を通してICカードチップ150にICカードコマンドを送信する。これにより、CPU121は、ICカードチップ150を利用して、ホスト機器220が要求するセキュリティ処理を実行する。CPU121は、特定のメモリカードコマンドの受信を契機に、ICカードチップ150に対する電源供給制御、クロック供給制御、ICカードコマンド送信、ICカードレスポンス受信を複数組み合わせることで操作することによって、セキュリティ処理を実行してもよい。また、CPU121は、ホスト機器220がMMC110へ電源供給を開始したのを契機として、セキュリティ処理を実行してもよい。セキュリティ処理の結果は、ICカードチップ150が出力するICカードレスポンスをベースにして構成され、MMC110内に保持される。MMC110がホスト機器220から特定のメモリカードコマンドを受信すると、CPU121はそれを契機として、セキュリティ処理の結果をホスト機器220に送信する。

【0046】図7は、ホスト機器220がMMC110にアクセスするときのフローチャートを表したものである。まず、ホスト機器220はMMC110を活性化するためにVCC1端子144に電源供給を開始する(701)。これを契機として、MMC110は、第1次ICカード初期化処理を実行する(702)。第1次ICカード初期化処理の詳細は後述する。次に、ホスト機器220はMMC110を初期化するためにCMD端子142を通してMMC110の初期化コマンドを送信する(703)。この初期化コマンドはMultiMedi

aCard仕様に準拠したものであり、複数種類ある。ホスト機器220は、MMC110を初期化するために、複数の初期化コマンドを送信する場合がある。MMC110が初期化コマンドを受信すると、MMC110はそれを処理する(704)。これを契機として、MMC110は、第2次ICカード初期化処理を実行する(705)。第2次ICカード初期化処理の詳細は後述する。ホスト機器220は、MMC110の初期化コマンドに対するメモリカードレスポンスを、CMD端子142を通して受信し、そのメモリカードレスポンスの内容からMMC110の初期化が完了したかを判定する。未完了ならば、再び初期化コマンドの送信をおこなう(703)。MMC110の初期化が完了したならば、ホスト機器220は、MultiMediaCard仕様に準拠した標準メモリカードコマンド(フラッシュメモリチップ130へアクセスするためのコマンド)や、上に述べたセキュリティ処理に関連した特定のメモリカードコマンド(ICカードチップ150へアクセスするためのコマンド)の送信を待機する状態に移る(707)。この待機状態では、ホスト機器220は標準メモリカードコマンドを送信することができる(708)。MMC110が標準メモリカードコマンドを受信したら、MMC110はそれを処理する(709)。処理が完了したら、ホスト機器220は、再び待機状態にもどる(707)。この待機状態では、ホスト機器220はセキュリティ処理要求ライトコマンドを送信することもできる(710)。セキュリティ処理要求ライトコマンドとは、上に述べたセキュリティ処理に関連した特定のメモリカードコマンドの1種であり、MMC110にセキュリティ処理を実行させるために処理要求を送信するメモリカードコマンドである。MMC110がセキュリティ処理要求ライトコマンドを受信したら、CPU121は、要求されたセキュリティ処理の内容を解釈し、セキュリティ処理をICカードコマンドの形式で記述する(711)。即ち、CPU121は、予め定められたルールに従って、ホスト機器230からの標準メモリカードコマンドを、ICカードチップ150が解釈可能な特定のメモリカードコマンドへ変換する。そして、その結果として得られたICカードコマンドをICカードチップ150に発行するなどして、要求されたセキュリティ処理を実行する(712)。処理が完了したら、ホスト機器220は、再び待機状態にもどる(707)。この待機状態では、ホスト機器220はセキュリティ処理結果リードコマンドを送信することもできる(713)。セキュリティ処理結果リードコマンドとは、上に述べたセキュリティ処理に関連した特定のメモリカードコマンドの1種であり、MMC110によるセキュリティ処理の実行結果を知るために処理結果を受信するメモリカードコマンドである。MMC110がセキュリティ処理結果リードコマンドを受信したら、CPU121は、IC

カードチップ150から受信したICカードレスポンスをベースに、ホスト機器220に送信すべきセキュリティ処理結果を構築する(714)。そして、ホスト機器220は、MMC110からセキュリティ処理結果を受信する。受信が完了したら、ホスト機器220は、再び待機状態にもどる(707)。なお、ステップ714は、ステップ712の中でおこなってもよい。

【0047】図7において、ステップ702およびステップ705で実行する第1次ICカード初期化処理および第2次ICカード初期化処理は、MMC110内でセキュリティ処理を実行するのに備えて、CPU121がICカードチップ150に対してアクセスする処理である。具体的には、ICカードチップ150の活性化や非活性化、ICカードチップ150のリセット、ICカードチップ150の環境設定を行う。環境設定とは、セキュリティ処理を実行するために必要な情報(例えば、使用可能な暗号アルゴリズムの情報、暗号計算に使用する秘密鍵や公開鍵に関する情報、個人認証に使用する認証データに関する情報など)をICカードチップ150から読み出したり、あるいはICカードチップ150に書き込んだりすることを意味する。ICカードチップ150の環境設定は、ICカードチップ150にICカードコマンドをN個(Nは正の整数)発行することによっておこなう。例えば、セッション鍵が3個必要ならば、ICカードコマンドを3回発行し、セッション鍵が2個必要ならば、ICカードコマンドを2回発行する。N個のICカードコマンドは、互いに相違するものであってもよいし、同一のものであってもよい。Nの値は固定されたものではなく、状況によってさまざまな値となる。以下、環境設定で発行するICカードコマンドを、設定コマンドと呼ぶ。また、この環境設定に基づいてセキュリティ処理を実行するICカードコマンドを、以下、セキュリティコマンドと呼ぶ。セキュリティコマンドの例としては、デジタル署名の計算、デジタル署名の検証、メッセージの暗号化、暗号化メッセージの復号、パスワードによる認証などをおこなうコマンドがある。

【0048】CPU121は、ICカードチップ150の環境設定の内容を自由に変更することができる。CPU121は、セキュリティ処理の内容や結果に応じてこれを変更してもよいし、ホスト機器からのメモリカードコマンドの受信を契機としてこれを変更してもよい。また、CPU121は、環境設定の内容を示した情報をフラッシュメモリチップ130にライトし、必要なときにフラッシュメモリチップ130からその情報をリードして使用することもできる。この情報は、図21においてICカード環境設定情報2112として示されている。これにより、MMC110が非活性化されてもその情報を保持することができ、MMC110が活性化されるたびにあらためて設定する手間を省くことができる。

【0049】第1次ICカード初期化処理および第2次

ICカード初期化処理は、ICカード制御パラメータA、B、Cに設定された値に基づいておこなわれる。また、CPU121は、ステップ712で実行するセキュリティ処理において、ICカード制御パラメータDに設定された値に基づいてICカードチップ150の活性化や非活性化を制御する。図8は、ICカード制御パラメータの種類と設定値、それに対応した処理の内容を表している。まず、パラメータAは、MMC110に電源が供給されたときに実行される第1次ICカード初期化処理に関するパラメータである。A=0のときは、CPU121はICカードチップ150にアクセスしない。A=1のときは、CPU121はICカードチップ150をコールドリセットする。A=2のときは、CPU121はICカードチップ150をコールドリセットした後でICカードチップ150の環境設定をおこなう。A=3のときは、CPU121はICカードチップ150をコールドリセットした後でICカードチップ150の環境設定をおこない、最後にICカードチップ150を非活性化する。A=0またはA=3のときは、第1次ICカード初期化処理のあとICカードチップ150が非活性状態となる。A=1またはA=2のときは、第1次ICカード初期化処理のあとICカードチップ150は活性状態となる。次に、パラメータBとCは、MMC110がMMC初期化コマンドを処理したときに実行される第2次ICカード初期化処理に関するパラメータである。B=0のときは、CPU121はICカードチップ150にアクセスしない。B=1かつC=1のときは、CPU121はICカードチップ150をリセット（コールドリセットまたはウォームリセット）する。B=1かつC=2のときは、CPU121はICカードチップ150をリセットした後でICカードチップ150の環境設定をおこなう。B=1かつC=3のときは、CPU121はICカードチップ150をリセットした後でICカードチップ150の環境設定をおこない、最後にICカードチップ150を非活性化する。B=2かつC=2のときは、CPU121はICカードチップ150の環境設定をおこなう。B=2かつC=3のときは、CPU121はICカードチップ150の環境設定をおこなった後にICカードチップ150を非活性化する。B=3のときは、ICカードチップ150が活性状態ならば、CPU121はICカードチップ150を非活性化する。最後に、パラメータDは、ホスト機器220から要求されたセキュリティ処理を実行したあとに、ICカードチップ150を非活性化するか否かを示すパラメータである。D=0のときは、セキュリティ処理の実行後に、CPU121はICカードチップ150を非活性化せず、活性状態に保つ。D=1のときは、セキュリティ処理の実行後に、CPU121はICカードチップ150を非活性化する。

【0050】CPU121は、ICカード制御パラメータ

A、B、C、Dの設定値を変更することができる。CPU121は、セキュリティ処理の内容や結果に応じてこれらの設定値を変更してもよいし、ホスト機器からのメモリカードコマンドの受信を契機としてこれらの設定値を変更してもよい。また、CPU121は、これらの設定値をフラッシュメモリチップ130にライトし、必要なときにフラッシュメモリチップ130からこれらの設定値をリードして使用することもできる。これらの設定値は、図21においてICカード制御パラメータ2111として示されている。これにより、MMC110が非活性化されてもこれらの設定値を保持することができ、MMC110が活性化されるたびにあらためて設定する手間を省くことができる。

【0051】図9は、第1次ICカード初期化処理のフローチャートを表している。初期化処理を開始する（901）と、まず、ICカード制御パラメータAが0かチェックする（902）。A=0ならばそのまま初期化処理は終了する（908）。A=0でないならばICカードチップ150をコールドリセットする（903）。次に、ICカード制御パラメータAが1かチェックする（904）。A=1ならば初期化処理は終了する（908）。A=1でないならばICカードチップ150の環境設定をおこなう（905）。次に、ICカード制御パラメータAが2かチェックする（906）。A=2ならば初期化処理は終了する（908）。A=2でないならばICカードチップ150を非活性化する（907）。そして、初期化処理は終了する（908）。

【0052】図10は、第2次ICカード初期化処理のフローチャートを表している。初期化処理を開始する（1001）と、まず、ICカード制御パラメータBが0かチェックする（1002）。B=0ならばそのまま初期化処理は終了する（1013）。B=0でないならばB=1かチェックする（1003）。B=1ならばICカード制御パラメータAが0または3かチェックする（1004）。Aが0または3ならば、ICカードチップ150をコールドリセットし（1005）、ステップ1007に移る。Aが1または2ならば、ICカードチップ150をウォームリセットし（1006）、ステップ1007に移る。ステップ1007では、ICカード制御パラメータCが1かチェックする。C=1ならば初期化処理は終了する（1013）。C=1でないならばステップ1009に移る。ステップ1003においてB=1でないならば、Bが2かチェックする（1008）。B=2ならばステップ1009に移る。B=2でないならば、ICカード制御パラメータAが0または3かチェックする（1011）。Aが0または3ならば初期化処理を終了する（1013）。Aが1または2ならば、ステップ1012に移る。ステップ1009ではICカードチップ150の環境設定をおこなう。そして、ICカード制御パラメータCが2かチェックする（10

10)。C=2ならば初期化処理を終了する(1013)。C=2でないならばステップ1012に移る。ステップ1012ではICカードチップ150を非活性化する。そして、初期化処理を終了する(1013)。

【0053】図11は、ICカードチップ150が非活性状態であるときに第1次ICカード初期化処理あるいは第2次ICカード初期化処理を実行した場合において、ICカードチップ150の外部端子の信号波形をシンプルに表したものである。図12は、ICカードチップ150が活性状態であるときに第2次ICカード初期化処理を実行した場合において、ICカードチップ150の外部端子の信号波形をシンプルに表したものである。図11と図12において、時間の方向は左から右にとっており、上の行から下の行に向かってVCC2端子151、RST端子152、CLK2端子153、I/O端子157で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準(Lレベル)を表す。図11において1102は図3に示したコールドリセットの信号波形を表す。図12において1202は図4に示したウォームリセットの信号波形を表す。図11と図12において、第1設定コマンド処理1104aと1204a、第2設定コマンド処理1104bと1204b、第N設定コマンド処理1104cと1204cは、それぞれ図5に示したICカードコマンド処理の信号波形を表す。ICカードチップ150の環境設定の信号波形1104と1204は、N個の設定コマンド処理の信号波形が連なって構成される。図11と図12において、1106と1206は、それぞれ図6に示した非活性化の信号波形を表す。図11と図12において、縦方向の破線1101、1103、1105、1107、1201、1203、1205、1207はそれぞれ特定の時刻を表す。1101はコールドリセット前の時刻、1201はウォームリセット前の時刻、1103はコールドリセット後から環境設定前の間にある時刻、1203はウォームリセット後から環境設定前の間にある時刻、1105と1205は環境設定後から非活性化前の間にある時刻、1107と1207は非活性化後の時刻である。

【0054】図11を参照して、第1次ICカード初期化処理実行時の信号波形を示す。ICカード制御パラメータAが0のときは、信号波形に変化はない。A=1のときは、時刻1101から時刻1103までの範囲の信号波形となる。A=2のときは、時刻1101から時刻1105までの範囲の信号波形となる。A=3のときは、時刻1101から時刻1107までの範囲の信号波形となる。

【0055】図11を参照して、ICカード制御パラメータAが0または3のときの、第2次ICカード初期化処理実行時の信号波形を示す。ICカード制御パラメータBが0のときは、信号波形に変化はない。B=1かつICカード制御パラメータC=1のときは、時刻110

1から時刻1103までの範囲の信号波形となる。B=1かつC=2のときは、時刻1101から時刻1105までの範囲の信号波形となる。B=1かつC=3のときは、時刻1101から時刻1107までの範囲の信号波形となる。

【0056】図12を参照して、ICカード制御パラメータAが1または2のときの、第2次ICカード初期化処理実行時の信号波形を示す。ICカード制御パラメータBが0のときは、信号波形に変化はない。B=1かつICカード制御パラメータC=1のときは、時刻1201から時刻1203までの範囲の信号波形となる。B=1かつC=2のときは、時刻1201から時刻1205までの範囲の信号波形となる。B=1かつC=3のときは、時刻1201から時刻1207までの範囲の信号波形となる。B=2かつC=2のときは、時刻1203から時刻1205までの範囲の信号波形となる。B=2かつC=3のときは、時刻1203から時刻1207までの範囲の信号波形となる。B=3のときは、時刻1205から時刻1207までの範囲の信号波形となる。

【0057】図13は、図7のステップ712において、CPU121が、ホスト機器220が要求したセキュリティ処理をICカードチップ150によって実行するときのフローチャートを表している。セキュリティ処理を開始する(1301)と、まずICカードチップ150が非活性状態かをチェックする(1302)。非活性状態ならば、ICカードチップ150をコールドリセットし(1303)、ステップ1306に移る。活性状態ならば、ステップ1304に移る。ステップ1304では、ICカードチップ150にICカードコマンドを発行する前にICカードチップ150を再リセットする必要があるかをチェックする。必要があるならば、ICカードチップ150をウォームリセットし(1305)、ステップ1306に移る。必要がないならば、ステップ1306に移る。ステップ1306では、ICカードチップ150の環境設定をおこなう必要があるかをチェックする。必要があるならば、ICカードチップ150の環境設定をおこない(1307)、ステップ1308に移る。必要がないならば、ステップ1308に移る。ステップ1308では、ICカードチップ150のCLK2端子に供給するクロック信号の周波数F2を設定する。そして、CPU121はICカードチップ150にセキュリティコマンドを発行し、ICカードチップ150はそれを処理する(1309)。セキュリティコマンドの処理時間は、クロック周波数F2に依存する。次に、ICカードチップ150が出力するICカードレスポンスにより、その処理が成功したかどうかを判定する(1310)。成功ならば、ステップ1311に移る。失敗ならば、ステップ1312に移る。ステップ1311では、ICカードチップ150に発行すべきセキュリティコマンドが全て完了したかをチェックする。発

行すべきセキュリティコマンドがまだあるならば、ステップ1304に移る。発行すべきセキュリティコマンドが全て完了したならば、ステップ1314に移る。ステップ1312では、失敗したセキュリティコマンドをリトライすることが可能かを判定する。リトライできるならば、リトライ設定をおこなない(1313)、ステップ1304に移る。リトライ設定とは、リトライすべきセキュリティコマンドやその関連データをCPU121が再度準備することである。リトライできないならステップ1314に移る。これは、ホスト機器220が要求したセキュリティ処理が失敗したことを意味する。ステップ1314では、ICカード制御パラメータDをチェックする。D=1ならば、ICカードチップ150を非活性化して(1315)、セキュリティ処理を終了する(1316)。D=1でないならば、ICカードチップ150を活性状態に保ったままセキュリティ処理を終了する(1316)。図13のフローチャートにおいては、クロック周波数F2を、ステップ1309で発行するセキュリティコマンドの種類によって変えることができるように、ステップ1308をステップ1309の直前に位置させたが、ステップ1308はそれ以外の位置にあってもよい。

【0058】従来のICカードへの攻撃法を有効にしている要因のひとつとして、ICカードの駆動クロックが外部の接続装置から直接供給されることがあげられる。駆動クロックが接続装置の制御下にあるため、タイミング解析や電力差分析においては、電気信号の測定においてICカード内部処理のタイミングの獲得が容易になる。一方、故障利用解析においては、異常な駆動クロックの供給による演算エラーの発生が容易になる。これに対し、本発明によれば、MMC110内部でICカードチップ150によりセキュリティ処理を実行するとき、ホスト機器220はICカードチップ150の駆動クロックを直接供給できない。CPU121は、ICカードチップ150へ供給するクロックの周波数F2を自由に設定することができる。これにより、ホスト機器220の要求する処理性能に柔軟に対応したセキュリティ処理が実現できる。ホスト機器220が高速なセキュリティ処理を要求するならば周波数F2を高く設定し、低い消費電力を要求するならば周波数F2を低く設定したり、クロックを適度に停止させればよい。また、CPU121は、周波数F2だけでなくクロックの供給開始タイミング、供給停止タイミングを自由に設定できる。これらをランダムに変化させることにより、ICカードチップ150に対するタイミング解析、電力差分析、故障利用解析と呼ばれる攻撃法を困難にすることができる。タイミング解析は、攻撃者が暗号処理1回の処理時間を正確に計測可能であることを仮定しているため、その対策としては、攻撃者が処理時間計測を正確に行えないようにすることが有効である。本発明によりタイミング解析

が困難になる理由は、ICカードチップ150がICカードコマンドを処理している時間の長さをホスト機器220が正確に計測できないためである。電力差分析の対策としては、処理の実行タイミングや順序に関する情報を外部から検出不可能にすることが有効である。本発明により電力差分析が困難になる理由は、ICカードコマンドが発行された時刻、発行されたICカードコマンドの内容、発行されたICカードコマンドの順序(ICカードコマンドを複数組み合わせるセキュリティ処理を実行する場合)の検出がホスト機器220にとって困難になるためである。故障利用解析の対策としては、ICカードにクロックや電圧や温度等の動作環境検知回路を搭載し、異常を検出したならば処理を停止あるいは使用不能にするという方法が有効である。本発明により故障利用解析が困難になる理由は、CLK2制御回路127がICカードチップ150に異常な駆動クロックを供給しないことが、ホスト機器220がICカードチップ150に演算エラーを発生させるのを防止するからである。

【0059】CPU121は、ICカードチップ150に供給するクロックの周波数F2、供給開始タイミング、供給停止タイミングの設定値を、セキュリティ処理の内容や結果に応じて変更してもよいし、ホスト機器からのメモリカードコマンドの受信を契機として変更してもよい。また、CPU121は、これらの設定値をフラッシュメモリチップ130にライトし、必要なときにフラッシュメモリチップ130からこれらの設定値をリードして使用することもできる。これらの設定値は、図21においてCLK2設定情報2113として示されている。これにより、MMC110が非活性化されてもこれらの設定値を保持することができ、MMC110が活性化されるたびにあらためて設定する手間を省くことができる。

【0060】図14は、ホスト機器220がセキュリティ処理要求ライトコマンドをMMC110に発行してから、ICカードチップ150でセキュリティ処理が実行されるまでの過程(図7のステップ710~712)において、MMC110およびICカードチップ150の外部端子の信号波形、CPU121によるフラッシュメモリチップ130へのアクセスをシンプルに表したものである。図14において、時間の方向は左から右にとる。一番上の行はフラッシュメモリチップ130へのアクセス内容である。上から二行目の行から下の行に向かって、VCC1端子144、CMD端子142、CLK1端子145、DAT端子147、VCC2端子151、RST端子152、CLK2端子153、I/O端子157で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準(Lレベル)を表す。図14を参照して、ホスト機器220がセキュリティ処理要求ライトコマンドをMMC110に発行してから、ICカー

ドチップ150でセキュリティ処理が実行されるまでの過程を説明する。まず、ホスト機器220はCMD端子142にセキュリティ処理要求ライトコマンドを送信する(1401)。次に、ホスト機器220はCMD端子142からセキュリティ処理要求ライトコマンドのレスポンスを受信する(1402)。このレスポンスは、MMC110がコマンドを受信したことをホスト機器220に伝えるものであり、セキュリティ処理の実行結果ではない。次に、ホスト機器220はDAT端子147にセキュリティ処理要求を送信する(1403)。セキュリティ処理要求とは、セキュリティ処理の内容や処理すべきデータを含むホストデータである。次に、MMC110はDAT端子147をLレベルにセットする(1404)。MMC110は、これによりビジー状態であることをホスト機器220に示す。次に、CPU121は、ホスト機器220から受信したセキュリティ処理要求をフラッシュメモリチップ130にライトするコマンドを発行する(1405)。セキュリティ処理要求をフラッシュメモリチップ130にライトすることにより、CPU121がセキュリティ処理要求をICカードコマンド形式で記述する処理(図7のステップ711)において、CPU121内部のワークメモリの消費量を節約できる。これは、セキュリティ処理要求のデータサイズが大きいときに有効である。なお、フラッシュメモリチップ130にライトされたセキュリティ処理要求は、図21においてセキュリティ処理バッファ領域2114に格納される。また、ライトコマンド発行1405は必須な操作ではない。ライト処理期間1406は、フラッシュメモリチップ130がセキュリティ処理要求のライト処理を実行している期間を表す。セキュリティ処理1407はICカードチップ150によるセキュリティ処理の信号波形を表す。この信号波形は図13のフローチャートの遷移過程に依存する。セキュリティ処理1407は、ライト処理期間1406とオーバーラップさせることができる。一般にフラッシュメモリチップ130のライト処理期間1406はミリ秒のオーダーであるため、セキュリティ処理1407とオーバーラップさせることは、セキュリティ処理の全体的な処理時間の短縮にとって有効である。リード/ライト1408は、セキュリティ処理1407の実行中に、フラッシュメモリチップ130からセキュリティ処理要求をリードしたり、ICカードチップ150が出力した計算結果をフラッシュメモリチップ130にライトするアクセスを示している。このアクセスにより、CPU121内部のワークメモリの消費量を節約できる。これは、セキュリティ処理要求やセキュリティ処理結果のデータサイズが大きいときに有効である。リード/ライト1408は必須ではない。セキュリティ処理1407が完了したら、MMC110はDAT端子147をHレベルにセットする(1409)。MMC110は、これによりセキュリティ処理が完了した

ことをホスト機器220に示す。

【0061】図15は、図14におけるセキュリティ処理1407の信号波形の一例を表したものである。図15において、時間の方向は左から右にとる。一番上の行はフラッシュメモリチップ130へのアクセス内容である。上から二行目の行から下の行に向かって、VCC2端子151、RST端子152、CLK2端子153、I/O端子157で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準(Lレベル)を表す。1501は図3に示したコールドリセットの信号波形を表し、1504は図4に示したウォームリセットの信号波形を表し、1502および1505は図11(あるいは図12)に示した環境設定の信号波形を表し、1503および1506および1507は図5に示したICカードコマンド処理の信号波形を表し、1508は図6に示した非活性化の信号波形を表す。ICカードチップ150の外部端子において図15に示した信号波形が観測されるのは、図13のフローチャートが1301、1302、1303、1306、1307、1308、1309、1310、1311、1304、1305、1306、1307、1308、1309、1310、1311、1304、1306、1308、1309、1310、1311、1314、1315、1316の順で遷移するときである。図15を参照して、図14のセキュリティ処理1407の実行中におけるCPU121によるフラッシュメモリチップ130へのアクセス(リード/ライト1408)を説明する。このアクセスには、図21におけるセキュリティ処理バッファ領域2114を使用する。リード1509、1511、1512は、それぞれ、セキュリティコマンド処理1503、1506、1507においてICカードチップ150に送信するICカードコマンドを構築するために必要なデータを、フラッシュメモリチップ130からリードするアクセスである。ライト1510は、セキュリティコマンド処理1503においてICカードチップ150が出力した計算結果を、フラッシュメモリチップ130にライトするアクセスである。ライト1513は、セキュリティコマンド処理1506および1507においてICカードチップ150が出力した計算結果を、フラッシュメモリチップ130にまとめてライトするアクセスである。リード1509、1511、1512は、それぞれ、セキュリティコマンド処理1503、1506、1507以前のICカードチップ150へのアクセスとオーバーラップさせることができる。ライト1510、1513は、それぞれ、セキュリティコマンド処理1503、1507以後のICカードチップ150へのアクセスとオーバーラップさせることができる。これらのオーバーラップは、セキュリティ処理の全体的な処理時間の短縮にとって有効である。さらに、フラッシュメモリチップ130のライト単位が大きい場合は、ライト1513のように

複数の計算結果をまとめてライトすることができる。これは、フラッシュメモリチップ130へのライト回数を削減し、フラッシュメモリチップ130の劣化を遅らせる効果がある。なお、ライト1510、1513でフラッシュメモリチップ130にライトする内容は、ICカードチップ150が出力した計算結果そのものに限定されず、図7のステップ715でホスト機器220に返すセキュリティ処理結果またはその一部であってもよい。この場合、図7のステップ714またはその一部は、ステップ712の中で実行されることになる。

【0062】図16は、ホスト機器220がセキュリティ処理結果リードコマンドをMMC110に発行してから、MMC110がセキュリティ処理結果を出力するまでの過程（図7のステップ713～715）において、MMC110の外部端子の信号波形、CPU121によるフラッシュメモリチップ130へのアクセスをシンプルに表したものである。図16において、時間の方向は左から右にとる。一番上の行はフラッシュメモリチップ130へのアクセス内容である。上から二行目の行から下の行に向かって、VCC1端子144、CMD端子142、CLK1端子145、DAT端子147で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準（Lレベル）を表す。図16を参照して、ホスト機器220がセキュリティ処理結果リードコマンドをMMC110に発行してから、MMC110がセキュリティ処理結果を出力するまでの過程を説明する。まず、ホスト機器220はCMD端子142にセキュリティ処理結果リードコマンドを送信する（1601）。次に、ホスト機器220はCMD端子142からセキュリティ処理結果リードコマンドのレスポンスを受信する（1602）。このレスポンスは、MMC110がコマンドを受信したことをホスト機器220に伝えるものであり、セキュリティ処理結果ではない。次に、MMC110はDAT端子147をLレベルにセットする（1603）。MMC110は、これによりビジー状態であることをホスト機器220に示す。次に、CPU121は、フラッシュメモリチップ130のセキュリティ処理バッファ領域（図21の2114）から、ICカードチップ150が出力した計算結果をリードする（1604）。CPU121は、これをもとにセキュリティ処理結果を構築し、MMC110がDAT端子147にセキュリティ処理結果を出力する（1605）。なお、図7のステップ714またはその一部が、ステップ712の中で実行されている場合、ステップ1604ではフラッシュメモリチップ130のセキュリティ処理バッファ領域（図21の2114）からセキュリティ処理結果またはその一部をリードする。なお、フラッシュメモリチップ130のセキュリティ処理バッファ領域（図21の2114）を利用しないでセキュリティ処理結果を構築する場合、ステップ1604は必要ない。

【0063】MMC110の製造者や管理者は、セキュリティシステムのユーザにMMC110を提供する前やそのユーザが所有するMMC110に問題が発生した時に、MMC110に内蔵されたICカードチップ150に様々な初期データを書きこんだり、ICカードチップ150のテストをおこなったりする必要がある。MMC110の製造者や管理者によるこれらの操作の利便性を高めるために、MMC110は、ICカードチップ150の外部端子をMMC外部端子140に割りつけるインタフェース機能を持つ。これにより、図3～図6で示したようなICカードチップ150へのアクセス信号を、MMC外部端子140から直接受信できる。このようなMMC110の動作モードを、MultiMedia Card仕様に準拠した動作モードと区別して、以下、インタフェース直通モードと呼ぶ。

【0064】インタフェース直通モードについて詳細に説明する。図17は、ICカードチップ150の外部端子をMMC外部端子140に割りつけるときの対応関係の一例を表している。この例では、RST端子152をCS端子141に割り付け、GND2端子155をGND1端子143、146に割り付け、VCC2端子151をVCC1端子144に割り付け、CLK2端子153をCLK1端子145に割り付け、I/O端子157をDAT端子147に割り付ける。このとき、CS端子141とCLK1端子145は入力端子、DAT端子147は入出力端子として機能する。

【0065】MMC110は、特定のメモリカードコマンドを受信すると、動作モードをインタフェース直通モードへ移したり、インタフェース直通モードからMultiMedia Card仕様に準拠した動作モードに戻ることができる。以下、動作モードをインタフェース直通モードへ移すメモリカードコマンドを直通化コマンド、動作モードをインタフェース直通モードから通常の状態に戻すメモリカードコマンドを復帰コマンドと呼ぶ。図1を参照して、MMC I/F制御回路123は、VCC2制御回路126、CLK2制御回路127、ICカードI/F制御回路128と接続されており、MMC110がホスト機器220から直通化コマンドを受信すると、CPU121の指示により図17で示した端子割り付けをおこなう。MMC110がホスト機器220から復帰コマンドを受信すると、CPU121の指示により図17で示した端子割り付けを解除し、MMC110はMultiMedia Card仕様に準拠した動作モードに戻る。

【0066】インタフェース直通モードでは、ホスト機器220がICカードチップ150に直接アクセスできるため、セキュリティの観点からインタフェース直通モードを利用できるのは限られた者だけに必要がある。そこで、直通化コマンドの発行には、一般のユーザに知られないパスワードの送信を必要とする。正しいパ

スワードが入力されないとインタフェース直通モードは利用できない。

【0067】図18は、ホスト機器220が、MMC110の動作モードをMultiMediaCard仕様に準拠した動作モードからインタフェース直通モードに移し、ICカードチップ150に直接アクセスし、その後、MMC110の動作モードを再びMultiMediaCard仕様に準拠した動作モードに戻すまでの処理のフローチャートを表している。ホスト機器220は処理を開始し(1801)、まずMMC110に直通化コマンドを発行する(1802)。MMC110は、直通化コマンドで送信されたパスワードが正しいかチェックする(1803)。正しければステップ1804に移り、間違っていれば処理は終了する(1810)。ステップ1804では、CPU121は、ICカードチップ150をコールドリセットする。そして、図17で示した端子割り付けをおこないインタフェースを直通化する(1805)。この時点から、ホスト機器220はICカードチップ150に直接アクセスする(1806)。ホスト機器220がICカードチップ150への直接アクセスを終了し、MMC110の動作モードを再びMultiMediaCard仕様に準拠した動作モードに戻すときは、MMC110に復帰コマンドを発行する(1807)。すると、CPU121は図17で示した端子割り付けを解除し、MMC110はMultiMediaCard仕様に準拠した動作モードに戻る(1808)。そして、CPU121は、ICカードチップ150を非活性化化する(1809)。以上で、処理は終了する(1810)。

【0068】図19は、図18のステップ1801~1806の過程において、MMC110およびICカードチップ150の外部端子の信号波形をシンプルに表したものである。図19において、時間の方向は左から右にとる。上の行から下の行に向かって、VCC1端子144、CMD端子142、CLK1端子145、DAT端子147、VCC2端子151、RST端子152、CLK2端子153、I/O端子157で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準(Lレベル)を表す。1905は、図3のコールドリセットの信号波形を示す。モード移行時刻1906は、動作モードがインタフェース直通モードに移る時刻を表す。

【0069】図19を参照して、ホスト機器220がMMC110の動作モードをMultiMediaCard仕様に準拠した動作モードからインタフェース直通モードに移しICカードチップ150に直接アクセスする過程を説明する。なお、MMC110のVCC1端子144には3V(VCC2端子151の標準電圧)が供給されている。ホスト機器220がCMD端子142に直通化コマンドを入力すると(1901)、CMD端子1

42から直通化コマンドのレスポンスが出力される(1902)。このレスポンスは、MMC110がコマンドを受信したことをホスト機器220に伝えるものである。次に、ホスト機器220はDAT端子147にパスワードを入力する(1903)。パスワード入力後、MMC110はDAT端子147にLレベルを出力し(1904)、ビジー状態であることをホスト機器220に示す。ビジー状態の間に、CPU121は、ICカードチップ150をコールドリセットする(1905)。そして、モード移行時刻1906において、動作モードをインタフェース直通モードに移す。このときに、DAT端子147はLレベルからハイレベル状態になる。これにより、ホスト機器220はビジー状態の解除を知ることができる。この時点から、ホスト機器220はICカードチップ150に直接アクセスする。例えば、CLK1端子145にクロックを供給すると(1907)、CLK2端子153にそのクロックが供給される(1908)。また、DAT端子147にICカードコマンドを送信すると(1909)、I/O端子157にそのICカードコマンドが送信される(1910)。

【0070】図20は、図18のステップ1807~1810の過程において、MMC110およびICカードチップ150の外部端子の信号波形をシンプルに表したものである。図20において、時間の方向は左から右にとる。上の行から下の行に向かって、VCC1端子144、CMD端子142、CLK1端子145、DAT端子147、VCC2端子151、RST端子152、CLK2端子153、I/O端子157で観測される信号を表す。また、横方向の破線はそれぞれの信号の基準(Lレベル)を表す。モード復帰時刻2003は、動作モードがインタフェース直通モードからMultiMediaCard仕様に準拠した動作モードに戻る時刻を表す。2004は、図6の非活性化の信号波形を示す。

【0071】図20を参照して、ホスト機器220がMMC110の動作モードをインタフェース直通モードからMultiMediaCard仕様に準拠した動作モードに戻す過程を説明する。なお、MMC110のVCC1端子144には3V(VCC2端子151の標準電圧)が供給されている。ホスト機器220がCMD端子142に復帰コマンドを入力すると(2001)、CMD端子142から復帰コマンドのレスポンスが出力される(2002)。このレスポンスは、MMC110がコマンドを受信したことをホスト機器220に伝えるものである。そして、モード復帰時刻2003において、MMC110はDAT端子147にLレベルを出力してビジー状態であることをホスト機器220に示し、それと同時に動作モードをMultiMediaCard仕様に準拠した動作モードに戻る。ビジー状態の間に、CPU121は、ICカードチップ150を非活性化する(2004)。そして、MMC110は、DAT端子1

47をハイインピーダンス状態にし(2005)、復帰
コマンドの処理が完了したことをホスト機器220に示
す。これ以後、ホスト機器220はICカードチップ1
50に直接アクセスできない。ホスト機器220が、C
LK1端子145にクロックを供給しながらCMD端子
142に何らかのメモリカードコマンドを送信した場
合、ICカードチップ150にそのクロック信号(20
06)は伝わらない。2001や2002においてホス
ト機器220がCLK1端子145に供給するクロック
信号は、ICカードチップ150のCLK2端子153
にも伝わるが、DAT端子147がハイインピーダンス
状態であるため、ICカードチップ150がICカード
コマンドを誤って認識することはない。

【0072】図21において、セキュリティ処理ステ
ータス領域2116には、ICカードチップ150による
セキュリティ処理の進捗状況を示す情報を格納する。C
PU121は、この情報をセキュリティ処理の実行中に
更新することができる。例えば、セキュリティ処理の途
中でMMC110への電源供給が停止した場合、電源供
給再開時にCPU121がこの情報をリードして参照す
れば、セキュリティ処理を中断した段階から再開する
ことができる。

【0073】本発明の実施形態によれば、メモリカード
外部からICカードチップの駆動クロックを直接供給し
ないため、ICカードチップの処理時間を正確に計測で
きず、また、処理の実行タイミングや順序の検出が困難
になる。さらに、異常な駆動クロックを供給することが
できず、演算エラーを発生させるのが困難になる。した
がって、タイミング解析、電力差分解析、故障利用解析
攻撃法に対するセキュリティが向上する。

【0074】本発明の実施形態によれば、メモリカード
外部からICカードチップの制御方式を自由に設定で
きる。例えば、高速処理が要求されるならば、ICカード
チップの駆動クロックの周波数を高くした制御方式を設
定し、低消費電力が要求されるならば、ICカードチ
ップの駆動クロックの周波数を低くしたり、ICカードチ
ップの駆動クロックを適度に停止させる制御方式を設定
することができる。したがって、セキュリティシステム
の要求する処理性能に柔軟に対応したセキュリティ処理
が実現できる。

【0075】本発明によれば、ICカードチップによる
セキュリティ処理に必要なデータや、ICカードチップ
を管理するための情報を、フラッシュメモリに保持する
ことができる。したがって、セキュリティ処理の利便性
を向上させることができる。

【0076】本発明の実施形態によれば、MMCの製造
者や管理者が、MMC内部のICカードチップに直接ア
クセスすることができる。したがって、MMC内部のI
Cカードチップの初期化やメンテナンスを、従来のIC
カードと同様な方法で実現できる。

【0077】本発明の実施形態によれば、フラッシュメ
モリチップを備えたMMCに、セキュリティ機能を追加
する場合、セキュリティ評価機関の認証を予め受けたI
Cカードチップ追加搭載することによって、セキュリテ
ィ評価機関によるMMCの認証が不要となるため、MM
Cの開発期間又は製造期間が短縮する。

【0078】本発明を適用したMMC110は、高度な
情報セキュリティが必要とされる銀行取引、クレジット
決済、コンテンツ配信など多様な電子商取引サービスに
適用することができる。ユーザにとっては、これらの多
様なサービス毎に異なるMMC110を持つよりも、1
枚で複数のサービスが利用できるMMC110を持つほ
うが便利である。1枚のMMC110を複数のサービス
に利用することを考えたとき、一般にサービス毎にその
事業者は異なるため、MMC110に内蔵されたICカ
ードチップ150の内部では、複数のサービス事業者が
各自のサービス用に設計開発したセキュリティ処理プロ
グラムが共存することになる(以下、ICカードチップ
150内のセキュリティ処理プログラムをICカードア
プリケーションと呼ぶ)。したがって、ICカードチ
ップ150には、サービスを利用するユーザが複数のIC
カードアプリケーションから使用したいものを選択でき
る機能(アプリケーション選択機能)や、複数のICカ
ードアプリケーションが互いに干渉して誤動作を起こし
たり秘密情報が解読されたりしないようにする機能(フ
ァイアウォール機能)が必要となる。そこで、ICカ
ードチップ150内に、アプリケーション選択機能やファ
イアウォール機能を持つオペレーティングシステム(以
下、ICカードOSと呼ぶ。)を設置し、各ICカード
アプリケーションは、その上で実行可能なプログラムと
して実装する。ICカードOSとしては、MULTOS
(MULTOSはMondex International Limitedの登録商標である。)や、Jav
aCardの仮想マシン(JavaCardはSun
Microsystems株式会社の登録商標であ
る。)などを適用することが望ましい。

【0079】ICカードOS上で実行可能なプログラム
は一般にICカード内のEEPROMに格納されるが、
本発明では、ICカードアプリケーションのプログラム
はICカードチップ150内のEEPROM162に格
納されるだけでなく、プログラムの一部がフラッシュメ
モリチップ130にも格納される。これにより、1つの
ICカードアプリケーションが消費するEEPROM1
62の量を削減することができ、ICカードチップ15
0に搭載可能なICカードアプリケーションの数を増や
すことができる。よって、ユーザは1枚のMMC110
でより多くのサービスを利用できる。

【0080】以下、図27～図37を参照しながら、I
Cカードアプリケーションのプログラムの一部をフラッ
シュメモリチップ130に格納したMMC110のアプ

リケーション管理機能について詳しく説明する。

【0081】図27は、ICカードチップ150の機能的構成を中心にして、MMC110の内部構成を示したものである。ICカードインタフェース171は、ICカードチップ150の外部端子（VCC2、RST、CLK2、GND2、I/O）、シリアルインタフェース161、ICカードコマンド／レスポンスの送受信を制御するプログラムをまとめて表したものである。ICカードOS172は、上で述べたICカードOSを表し、ICカードインタフェース171を通して外部から受信したICカードコマンドに応じて、ICカードアプリケーションの選択をおこなう。また、選択されたICカードアプリケーションへコマンドデータの引き渡しをおこない、選択されたICカードアプリケーションから受け取った処理結果データに基づいてICカードレスポンス作成し、ICカードインタフェース171を通して外部へ送信する。アプリケーションA173、アプリケーションB174、アプリケーションC175は、それぞれICカードOS172の上で実行可能なプログラムとして実装されたICカードアプリケーションであり、ユーザが利用したいサービスに応じてMMC110外部からメモリカードコマンドで選択することができる（例えば、銀行取引サービスを利用したい時はアプリケーションA173を、クレジット決済サービスを利用したい時はアプリケーションB174を、コンテンツ配信サービスを利用したい時はアプリケーションC175を選択するなど）。次に、アプリケーションA173内のコマンド解釈部181、モジュール実行部182、モジュールロード部183、モジュール管理情報184、モジュールロード領域185、認証部186およびフラッシュメモリチップ130内のコマンド処理モジュール187は、アプリケーションA173のプログラムの一部をフラッシュメモリチップ130に格納するという機能を実現するための機能ブロックやデータを表している。フラッシュメモリチップ130に格納されているアプリケーションA173のプログラムの一部をコマンド処理モジュール187と呼ぶ。コマンド処理モジュール187は、アプリケーションA173が選択された時に使用することができるICカードコマンド中の1つのコマンドを処理するプログラムモジュールであり、コマンド毎に用意されている。コマンド処理モジュール187はフラッシュメモリチップ130に複数格納することができる。コマンド処理モジュール187はフラッシュメモリチップ130に格納されているときは暗号化されており、コマンド処理の内容はアプリケーションA173に対応するサービスの事業者以外に知られないようになっており、アプリケーションA173内で暗号が解かれてモジュールロード領域185にロードされてはじめて実行が可能となる。各機能ブロックやデータがどのように作用するかについては、以下の電子取引サービスの具体

例を用いて後で詳しく説明する。

【0082】電子取引サービスの具体例として銀行取引サービスを示す。図28～図30を用いてそのサービスを詳細に説明する。

05 【0083】図28は、MMCとのインタフェース持つ携帯端末を利用してユーザが銀行取引サービスを実行するシステムの構成を表している。ホスト機器220は、MMCとのインタフェースを持つ携帯端末であり、ホストインタフェース223を通してMMC110にコマンドによってアクセスする。銀行取引サーバ2830は、ユーザの銀行口座へのアクセスを提供し、ユーザからの指示に応じて銀行取引を実行する。ホスト機器220は通信手段2813を持つ。通信手段2813は、ホスト機器220がネットワーク2820を通して銀行取引サーバ2830に接続して情報交換をおこなう際の通信処理をおこなう。ホスト機器220は、CPU2811をもつ。CPU2811は、ホストインタフェース223や通信手段2813を制御し、MMC110や銀行取引サーバ2830との間で情報交換をおこなう。ホスト機器220は情報表示手段2814を持つ。情報表示手段2814は、CPU2811により制御され、銀行取引に関する情報をユーザに表示する。ホスト機器220はユーザ入力手段2812を持つ。ユーザ入力手段2812から入力されたデータはCPU2811で処理される。ユーザは、ユーザ入力手段2812を用いてユーザ認証のための暗証番号を入力したり、所望の銀行取引（残高照会、振込みなど）を指示する。

【0084】図29および図30は、図28のシステムにおいて銀行取引サービスを実行するときの手順を示したフローチャートである。ここでは簡単のため、エラー発生時の処理フローを省略する。まず、ユーザは、MMC110を利用したセキュリティ処理をおこなうため、第1PIN（第1の暗証番号）によってMMC110から認証されなければならない。ホスト機器220は、ユーザがユーザ入力手段2812から入力した第1PINをMMC110へ送信するため、第1PIN検証コマンドを発行する（2901）。MMC110は、内部のICカードチップ150に第1PIN検証のためのICカードコマンドを送信し、受信した第1PINが正しいかを検証する（2902）。検証が成功すると、銀行取引サービスに対応したICカードアプリケーションによるセキュリティ処理の利用が許可される。そこで、ホスト機器220は、MMC110に銀行取引アプリケーション選択コマンドを発行する（2903）。MMC110は、内部のICカードチップ150にアプリケーション選択のためのICカードコマンドを送信し、銀行取引サービスに対応したICカードアプリケーションを利用可能な状態にする（2904）。次に、ホスト機器220は、銀行取引サーバ2830に銀行取引開始要求のメッセージを送信する（2905）。銀行取引サーバ283

0 は、認証局が発行したサーバ証明書（サーバ公開鍵を含む）をホスト機器220に送信する（2906）。ホスト機器220は、サーバ証明書を検証するため、サーバ証明書のハッシュ値を計算する（2907）。そして、ホスト機器220は、MMC110に署名検証コマンドを発行して、そのハッシュ値とサーバ証明書につけられた認証局による署名を送信する（2908）。MMC110は、内部のICカードチップ150に署名検証のためのICカードコマンドを送信し、受信した署名が正しいかを認証局の公開鍵を用いて検証する（2909）。次に、ホスト機器220は、MMC110に乱数発生コマンドを発行する（2910）。MMC110は、内部のICカードチップ150に乱数発生のためのICカードコマンドを送信する。ICカードチップ150は乱数を発生し、それをステップ2914のために一時的に保持する（2911）。そして、ホスト機器220は、サーバ証明書からサーバ公開鍵を抽出し（2912）、MMC110に暗号化コマンドを発行してサーバ公開鍵を渡す（2913）。MMC110は、内部のICカードチップ150に暗号化のためのICカードコマンドを送信し、ステップ2911で発生した乱数をサーバ公開鍵で暗号化させ、ホスト機器220に返す（2914）。ホスト機器220は、暗号化した乱数と、認証局が発行したユーザ証明書（ユーザ公開鍵を含む）を銀行取引サーバ2830に送信する（2915）。銀行取引サーバ2830は、サーバ秘密鍵で暗号化乱数を復号して乱数を取得し、ユーザ証明書を認証局の公開鍵を用いて検証し、ユーザ証明書の検証が成功すれば、そこからユーザ公開鍵を抽出し、取得した乱数をユーザ公開鍵で暗号化する（2916）。そして図30に移って、銀行取引サーバ2830は、ユーザ公開鍵で暗号化した乱数をホスト機器220に送信する（3001）。ホスト機器220はこの暗号化乱数を受信する（3002）。そして、ホスト機器220は、銀行取引サーバ2830が本物かどうかを知るため暗号化乱数から乱数を復元できるかを調べる。そこで、MMC110に復号・比較コマンドを発行して、暗号化乱数を送信する（3003）。MMC110は、内部のICカードチップ150に復号・比較のためのICカードコマンドを送信する。ICカードチップ150はユーザ秘密鍵で暗号化乱数を復号し、ステップ2911で発生した乱数と比較し、一致したかどうかの結果を返す（3004）。ホスト機器220は、MMC110から乱数が一致したことを示す応答を受け取ると、情報表示手段2814を用いてユーザに銀行との取引内容を指示するよう求める。ユーザはユーザ入力手段2812から取引内容を指示する。そして、ホスト機器220は、指示された取引内容（残高照会、振込みなど）を銀行取引サーバ2830に送信する（3005）。銀行取引サーバ2830は、受信した取引内容が本当にユーザからのものかを確認するため、取

引内容を記した銀行取引契約書を作成し、作成した銀行取引契約書へのユーザの署名を求めるため、作成した銀行取引契約書をホスト機器220に送信する（3007）。ホスト機器220は、受信した銀行取引契約書を情報表示手段2814に表示する（3008）。そしてユーザに内容確認および署名を求める。ユーザは、内容が正しいことを確認したならば、MMC110を利用してユーザ秘密鍵による銀行取引契約書への電子的な署名をおこなう。MMC110は署名処理の実行を許可する前に、第2のPINによって再度ユーザを認証する。ユーザはユーザ入力手段2812から第2PINを入力する。ホスト機器220は、入力された第2PINをMMC110へ送信するため、第2PIN検証コマンドを発行する（3009）。MMC110は、内部のICカードチップ150に第2PIN検証のためのICカードコマンドを送信し、受信した第2PINが正しいかを検証する（3010）。検証が成功すると、ユーザ秘密鍵による署名処理の利用が許可される。ホスト機器220は、銀行取引契約書への署名を作成するために銀行取引契約書のハッシュ値を計算する（3011）。そして、ホスト機器220は、MMC110に署名作成コマンドを発行して、そのハッシュ値を送信する（3012）。MMC110は、内部のICカードチップ150に署名作成のためのICカードコマンドを送信し、ハッシュ値とユーザ秘密鍵により署名を作成する（3013）。ホスト機器220は、作成した署名をつけた銀行取引契約書を銀行取引サーバ2830に送信する（3014）。銀行取引サーバ2830は、受信した銀行取引契約書につけられた署名を、図29のステップ2916で取得したユーザ公開鍵を用いて検証する（3015）。検証が成功すれば、その取引内容は本物のユーザが指示したものであることが証明され、銀行取引サーバ2830は、指示された銀行取引の処理を実行する（3016）。処理が完了したら、銀行取引サーバ2830は、完了通知のメッセージをホスト機器220に送信する（3017）。ホスト機器220は、完了通知のメッセージを受信して情報表示手段2814にそれを表示し、ユーザに取引の完了を通知する（3018）。以上が、銀行取引サービスの実行手順である。

【0085】図27において、例えば、アプリケーションA173が、以上の銀行取引サービスを実行するためのICカードアプリケーションであるとする。以下、銀行取引サービスを例にして、アプリケーションA173のプログラムの一部をフラッシュメモリチップ130に格納するという機能について具体的に説明していく。

【0086】図31は、図27のコマンド処理モジュール187がフラッシュメモリチップ130のどこに格納されるかを詳細に示したものである。アプリケーションA用コマンド処理モジュール群3111は、銀行取引サービスの実行手順においてMMC110内のICカード

チップ150に送信されるICカードコマンドを処理するためのコマンド処理モジュール187を、複数まとめたものである。図32は、アプリケーションA用コマンド処理モジュール群3111の内容を示している。銀行取引サービスでは7つのICカードコマンドが使用されるため、7つのコマンド処理モジュール187が含まれている。第1PIN検証処理モジュール3201は、図29のステップ2902で使用する。署名検証処理モジュール3202は、図29のステップ2909で使用する。乱数発生処理モジュール3203は、図29のステップ2911で使用する。暗号化処理モジュール3204は、図29のステップ2914で使用する。復号・比較処理モジュール3205は、図30のステップ3004で使用する。第2PIN検証処理モジュール3206は、図30のステップ3010で使用する。署名作成処理モジュール3207は、図30のステップ3013で使用する。これらのコマンド処理モジュール187は識別番号1から7によって管理され、アプリケーションA用コマンド処理モジュール群3111の中から必要なものを引き出すことができる。図31において、アプリケーションB用コマンド処理モジュール群3112、アプリケーションC用コマンド処理モジュール群3113も、アプリケーションA用コマンド処理モジュール群3111と同様に、それぞれのICカードアプリケーションに対応する電子取引サービスで使用されるICカードコマンドを処理するためのコマンド処理モジュール187を、複数まとめたものである。

【0087】コマンド処理モジュール187を使用するためにはICカードチップ150へロードする必要がある。このロード処理は、図13のステップ1306や1307で示した環境設定の一種である。そのため、アプリケーションA用コマンド処理モジュール群3111、アプリケーションB用コマンド処理モジュール群3112、アプリケーションC用コマンド処理モジュール群3113といったコマンド処理モジュール群は、図31のように、フラッシュメモリチップ130の管理領域2110におけるICカード環境設定情報領域2112に格納する。管理領域2110以外の領域（ホストデータ領域2115）に格納してもよいが、不正に改ざんされたりすることを防止するため、管理領域2110に格納するほうが望ましい。モジュールデータ領域3110は、各コマンド処理モジュール群を格納するために、ICカード環境設定情報領域2112内に用意された領域である。なお、図31では図21に示した領域のうち、説明に必要ないものを省略している。

【0088】次に、ICカードチップ150に登録されたアプリケーションA173の中にあるモジュールロード領域185について詳細に説明する。図33は、ICカードチップ150内のメモリ資源（ROM159、RAM160、EEPROM162）の詳細な内部構成を

示したものである。まず、ROM159には、ICカードOSプログラム3341、ICカードインタフェース制御プログラム3342が含まれる。ICカードOSプログラム3341は、ICカードOS172を機能させるためにCPU158で実行されるプログラムである。ICカードインタフェース制御プログラム3342は、ICカードインタフェース171を制御するためにCPU158で実行されるプログラムである。次に、RAM160には、ICカードOSワーク領域3351、アプリケーションワーク領域3352が含まれる。ICカードOSワーク領域3351は、CPU158がICカードOSプログラム3341を実行するときに使用するメインメモリである。アプリケーションワーク領域3352は、ICカードOS172によって選択されたICカードアプリケーション（173、174、175など）が使用するメインメモリである。次に、EEPROM162には、ICカードOS設定情報3310、アプリケーションA173、アプリケーションB174、アプリケーションC175などが含まれる。ICカードOS設定情報3310は、ICカードOS172が、状況に応じてICカードOSプログラム3341によって既定された機能を変化させたり、新しい機能を追加するために使用する書き換え可能な情報である。各ICカードアプリケーション（173、174、175など）は、さらに3つの構成要素からなる。図33では、アプリケーションA173についてそれらの構成要素（アプリケーションAメインプログラム3320、モジュールロード領域185、モジュール管理情報184）を示す。アプリケーションAメインプログラム3320は、アプリケーションA173の4つの機能ブロック（図27における、コマンド解釈部181、モジュール実行部182、モジュールロード部183、認証部186）を含み、ICカードOS172上で実行されるプログラムである。モジュールロード領域185は、アプリケーションA用コマンド処理モジュール群3111の7モジュール（3201～3207）から実行したいものをロードするための領域であり、1つ以上のモジュールを格納できるサイズが確保されている。例えば、3つの領域（第1領域3331、第2領域3332、第3領域3333）が用意されている。この場合、7モジュールのうち最大3モジュールまでロードすることができる。よって、EEPROM162において、アプリケーションA173に消費される領域サイズをできるだけ小さくしたいならば、モジュールロード領域185は、1モジュールを格納できるサイズを確保すればよい。次に、モジュール管理情報184は、各コマンド処理モジュール（3201～3207）をモジュールロード領域185にロードするときなどに使用されるデータであり、アプリケーションA173内でその内容を参照したり更新したりすることができる。また、MMC110内のコントローラチップ1

20がその内容を読み出すこともできる。さらに、アプリケーションA173の改訂などにおいて、そのコマンド処理モジュール(3201~3207)を更新する際には、コントローラチップ120がその内容を書き換えることもできる。

【0089】図35は、アプリケーションA173のモジュール管理情報184の内容を示している。モジュール管理情報184は、ロード管理情報3510とロード領域情報3520から構成される。ロード管理情報3510は、アプリケーションA用コマンド処理モジュール群3111の各モジュール(3201~3207)に関する情報であり、ロード領域情報3520は、モジュールロード領域185の各領域(3331~3333)に関する情報である。ロード管理情報3510は、各モジュールの識別番号(番号3511)とコマンド処理の内容(処理内容3512)との対応関係を示す情報を含む。コントローラチップ120がモジュールをロードする際には、番号3511をコマンド処理モジュール187に付してICカードチップ150にロードする。なお、処理内容3512に記載するデータとして、ホスト機器220から受信したセキュリティ処理要求コマンドのヘッダ部分(コマンドクラス番号や命令コード番号など)を利用すれば、コントローラチップ120によるモジュール選択がしやすい。ロード管理情報3510は、各モジュールのプログラムサイズ(サイズ3513)

(単位はバイト)を含み、モジュールロード領域185の各領域に格納可能なサイズかどうかを判定できるようにしている。ロード管理情報3510は、各モジュールのロード状態(状態3514)を含む。図35では、例えば、第1PIN検証処理モジュール3201がモジュールロード領域185の第2領域3332にロードされており、乱数発生処理モジュール3203はモジュールロード領域185にロードされていないことを示している。ロード管理情報3510は、その時点でロードされている各モジュールの改訂番号(使用版3515)を含む。ロード管理情報3510は、各モジュールの最新の改訂番号(最新版3516)を含む。モジュールの更新時にMMC110のホスト機器220から最新の改訂番号が知らされ、そのモジュールの最新版3516の値がその番号に更新される。その際、更新するモジュール以外のモジュールの最新版3516の値も同時に更新してもよい。ロード管理情報3510は、ロード条件3517を含む。ロード条件3517は、モジュールをロードするときに、ロードを許可するかどうかをアプリケーションA173が判定するための条件であり、その使用版3515や最新版3516の値に基づく。図35では、例えば、署名検証処理モジュール3202は改訂番号に関係なくロードを許可し、暗号化処理モジュール3204は最新版のみロードを許可し、復号・比較処理モジュール3205は使用版3515の値が1.0以上ならば

ロードを許可することを意味する。次に、ロード領域情報3520は、モジュールロード領域185の各領域(3331~3333)の容量サイズ3521(単位はバイト)を含む。これは、各領域(3331~3333

3)が、サイズ3521の値より大きなサイズのモジュールをロードできないことを示すための情報である。例えば、第2領域3332には320バイトの容量しかないため、復号・比較処理モジュール3205(サイズが388バイト)はロードすることが許可されない。ロード領域情報3520は、領域識別番号3522を含む。これは、コントローラチップ120が、コマンド処理モジュール187をロードする領域を指定するのに使用する。

【0090】図34は、コントローラチップ120によるコマンド処理モジュール187のロード、およびICカードチップ150によるセキュリティコマンドの実行の手順を示すフローチャートである。まず、コントローラチップ120はモジュール管理情報184を使用してモジュールをロードするかを判断する(3401)。このステップ3401は、図13ではステップ1306に相当する。コントローラチップ120が、必要なモジュールがすでにロードされていることを知っているならば、ステップ3412に移ってよい。これは、図13ではステップ1306から1308への遷移に相当する。一方、コントローラチップ120が、モジュール管理情報184を使用してモジュールをロードする必要があるかを判断したいならば、ステップ3402に遷移する。これは、図13ではステップ1307(環境設定の実行)への遷移に相当する。ステップ3402では、コントローラチップ120は、環境設定のためのICカードコマンドとして、モジュール管理情報184のリードコマンドをICカードチップ150に発行する。ICカードチップ150のICカードOS172は、アプリケーションA173にこのコマンドの処理権を渡す。アプリケーションA173は、コマンド解釈部181でこのコマンドを解釈し、ICカードOS172を通してモジュール管理情報184をコントローラチップ120へ送信する(3403)。コントローラチップ120は、モジュール管理情報184の内容を調べ、所望のコマンド処理モジュール187が、モジュールロード領域185にロードされているかを確認する(3404)。そして、存在するならばステップ3412に移り、存在しなければステップ3406に移る(3405)。コントローラチップ120は、フラッシュメモリチップ130のモジュールデータ領域3110から所望のコマンド処理モジュール187(ここでは3201から3407のいずれか)を読み出す(3406)。そして、コントローラチップ120は、環境設定のためのICカードコマンドとして、モジュールロードコマンドをICカードチップ150に発行し、読み出したモジュール、モジュール識別

番号（ここでは1～7の範囲）、ロード領域番号（ここでは1～3の範囲）を送信する（3407）。ICカードチップ150のICカードOS172は、アプリケーションA173にこのコマンドの処理権を渡す。アプリケーションA173は、コマンド解釈部181でこのコマンドを解釈し、認証部186において、受信したコマンド処理モジュール187が適正かを判定する（3408）。具体的には、コマンド処理モジュール187にかけられた暗号を解き、実行可能なデータ形式であるか、サイズが適当であるかなどを調べる。そのため、認証部186は、モジュールにかけられた暗号を解くためのモジュール鍵を持つ。望ましくは、モジュールの改ざん防止のために電子署名を適用する。すなわち、アプリケーションA173に固有の秘密鍵（アプリケーション秘密鍵）とそれに対応する公開鍵（アプリケーション公開鍵）を用意し、コマンド処理モジュール187、または、その暗号化前の実行可能形式データにアプリケーション秘密鍵によって電子署名を付けておき、認証部186でアプリケーション公開鍵によってその署名を検証するようにする。この場合、認証部186は、アプリケーション公開鍵も持つ。ステップ3408において、コマンド処理モジュール187が適正でないならば、ステップ3411に移り、ICカードチップ150は、ロード結果が失敗であることをコントローラチップ120に返す。コマンド処理モジュール187が適正ならば、アプリケーションA173のモジュールロード部183は、モジュールロード領域185の指定された領域（3331、3332、3333のいずれか）にライトし（3409）、モジュール管理情報184の状態3514を更新する（3410）。そして、ICカードチップ150は、ロード結果が成功であることをコントローラチップ120に返す（3411）。そして、コントローラチップ120は、ステップ3412～3416にわたって、アプリケーションA173によるセキュリティコマンド処理を実行する。これは、図13のステップ1309に相当する。まず、コントローラチップ120は、ICカードチップ150にセキュリティコマンドを発行する（3412）。セキュリティコマンドとは、図29におけるステップ2901、2908、2910、2913、図30におけるステップ3003、3009、3012のいずれかで発行するコマンドに応じて、MMC110内部で発行されるICカードコマンドである。ICカードチップ150内のアプリケーションA173のコマンド解釈部181は、モジュール管理情報184により、そのコマンドに対応するコマンド処理モジュール187がモジュールロード領域185に存在するかを調査する（3413）。存在しないならば、処理結果が失敗であることをコントローラチップ120に返し（3415）、ステップ3416に移る。存在するならば、モジュール実行部182は、モジュールロード領域185か

らそのコマンド処理モジュール187を読み、セキュリティコマンドを処理する（3414）。そして、その処理結果をコントローラチップ120に返す（3415）。コントローラチップ120は、その処理結果を受信する（3416）。以上が、コントローラチップ120によるコマンド処理モジュール187のロード、およびICカードチップ150によるセキュリティコマンドの実行手順である。

【0091】図36は、フラッシュメモリチップ130に格納されたアプリケーションA173のためのコマンド処理モジュール187を、ホスト機器220によって更新する手順を示すフローチャートである。ここで、ホスト機器220は、アプリケーションA173のモジュール更新することを許可されており、上述のアプリケーション秘密鍵を持っているものとする。また、この手順においてホスト機器220からMMC110に送信される新しいコマンド処理モジュールは、その識別番号、サイズ、改訂番号（それぞれ、図35における番号3511、サイズ3513、最新版3516に登録されるべき情報）が含まれている。また、この手順に先だって、ICカードチップ150では、アプリケーションA173が選択されているものとする。まず、ホスト機器220は暗号化乱数発生コマンドを発行する（3601）。MMC110は、ICカードチップ150のアプリケーションA173の認証部186において、乱数を発生させ（3602）、その乱数を上述のアプリケーション公開鍵で暗号化させ、暗号化乱数をホスト機器220に送信する（3603）。ホスト機器220は、アプリケーション秘密鍵で暗号化乱数を復号し、乱数を復元する（3604）。そして、ホスト機器220は、その乱数と新しいコマンド処理モジュールを連結したデータを作り、そのデータにアプリケーション秘密鍵で電子的に署名する（3605）。ホスト機器220は、MMC110に検証・ライトコマンドを発行して、その署名付き連結データを送信する（3606）。MMC110は、ICカードチップ150のアプリケーションA173の認証部186において、アプリケーション公開鍵で連結データの署名を検証させる（3607）。ステップ3608では、連結データ内の乱数がステップ3602で発生した乱数と一致するかを比較する。ステップ3607での検証が成功し、かつそれらの乱数が一致したならば、受信した新しいコマンド処理モジュールは正しいものであることが証明されたので、新しいコマンド処理モジュールを、アプリケーションA用コマンド処理モジュール群3111の中の同じ識別番号のコマンド処理モジュールに上書きする（3609）。さらに、ICカードチップ150にICカードコマンドを送ることによって、アプリケーションA173内のモジュール管理情報184の内容（図35における番号3511、サイズ3513、最新版3516に登録される情報）も更新する。そして、

ホスト機器220は更新処理を終了する(3610)。一方、ステップ3607での検証が失敗するか、または乱数が一致しないならば、コマンド処理モジュールやモジュール管理情報184の内容の更新は実行せず、ホスト機器220は更新処理を終了する(3610)。

【0092】図37は、図34や図36の手順で登場したアプリケーションに固有な3種類の鍵(モジュール鍵、アプリケーション秘密鍵、アプリケーション公開鍵)を説明したものである。モジュール鍵3701は、コマンド処理モジュールにかけられた暗号を解き、実行可能形式に復元するための対称鍵である。アプリケーション秘密鍵3702は、アプリケーション発行者が厳重に管理すべき秘密鍵であり、コマンド処理モジュールの更新手順において乱数の復号、署名の作成に使用される。アプリケーション公開鍵3703は、アプリケーション秘密鍵3702に対応する公開鍵であり、コマンド処理モジュールのロード手順・更新手順において乱数の暗号化、署名の検証に使用される。

【0093】上記では、コマンド処理モジュール187が1つのセキュリティコマンドを処理するプログラムモジュールであると説明されているが、複数のセキュリティコマンドを1つのコマンド処理モジュール187が処理してもよい。

【0094】本発明によれば、ICカードチップによるセキュリティ処理に必要なデータやICカードチップを管理するための情報をフラッシュメモリに保持し、ICカードチップ内のセキュリティ処理プログラムが消費するメモリ資源を節約し、より多くのプログラムを登録することができる。したがって、ユーザの利便性を向上させることができる。

【0095】尚、1つのMMC110が、複数のICカードチップ150を備えてもよい。複数のICカードチップ150の個々は、異なるアプリケーションプログラムを実行してもよい。例えば、3つのICカードチップ150のうちの第1のICカードチップ150が、アプリケーションプログラムA173を実行し、3つのICカードチップ150のうちの第2のICカードチップ150が、アプリケーションプログラムB174を実行し、3つのICカードチップ150のうちの第3のICカードチップ150が、アプリケーションプログラムC175を実行するのが好ましい。さらに、複数のICカードチップ150の個々は、異なる者によって認可又は発行されてもよい。例えば、第1のICカードチップ150は、アプリケーションプログラムA173を用いて銀行取引サービスを実現する例えば銀行によって認可又は発行され、第2のICカードチップ150は、アプリケーションプログラムB174を用いてクレジット決済サービスを実現する例えばクレジット会社によって認可又は発行されるのが好ましい。第3のICカードチップ150は、アプリケーションプログラムC175を用いてコンテンツ配信サービスを実現する例えばコンテン

ツプロバイダーによって認可又は発行されてもよいし、第3者(例えば、ICカードチップ150のセキュリティを認可又は保証する機関)によって認可又は発行されてもよい。この場合、銀行、クレジット会社、コンテンツプロバイダーや第3者によって認可又は発行された後、MMC110に搭載するのが好ましい。つまり、認可又は発行されたICカードチップ150を、MMC110上に固定し、ICカードチップ150内のインターフェース150~157と、ICカードI/F制御回路128、CLK2制御回路127、VOC2制御回路126、GND1とを電氣的に接続する。

【0096】これまでに述べたMMC110のセキュリティ処理においては、ICカードチップ150に送信するICカードコマンドによって扱うべきデータサイズが、ICカードチップ150内部で利用可能なワークメモリの空き容量を越えてしまうと、セキュリティ処理が実行不能になってしまう可能性がある。例えば、ワークメモリ(例えば、RAM160のICカードOSワーク領域3351やアプリケーションワーク領域3352等)の空き容量が100バイトのときに、ICカードコマンドにより200バイトの暗号文の復号が要求されても処理できない。このような問題を解決するために、MMC110は、ICカードチップ150が状況(例えば、ICカードコマンドによって要求される処理に必要なメモリ量やICカードコマンドによって処理すべきデータ量)に応じて外部に対して能動的に要求を出す機能を搭載する。以下、その機能について説明する。

【0097】上記の問題を解決するには、まず、ICカードチップ150は、セキュリティのためのコマンドを受け取り、その後処理すべきデータ量を算出し、その処理すべきデータ量がその時点でのワークメモリの空き容量より大きい又は以上であるかを判断するとよい。比較すべき空き容量は、コマンドを受け取ってからその都度計算してもよいし、常時計算しておいてレジスタに保持しておき、その値を利用してもよい。もしそれが空き容量より大きい又は以上でなければ(空き容量より小さい又は以下であれば)、コマンドで指示された処理を実行し、コントローラチップ120にその処理結果を応答として返す。一方、もしそれが空き容量より大きい又は以上であるならば、状況に応じた要求を含む応答をコントローラチップ120に返す。その要求の内容としては、例えば、「たったいま送信したデータをフラッシュメモリチップ130にライトし、128バイト単位に分割して送れ」とか、「ホスト機器220は、処理すべきデータを128バイト単位に分割して再送信せよ」といったものが考えられる。コントローラチップ120が、このような要求にしたがって適当な処理を実行すれば、ICカードチップ150によるセキュリティ処理が実行不能になることを防止することができる。

【0098】このような機能は、図23を用いて説明し

たコンテンツ配信の例にも適用することができる。暗号化されたコンテンツ2314をセッション鍵によって復号する手順2324において、コンテンツ2314のデータサイズがワークメモリの空き容量より大きい又は以上であったならば、ICカードチップ150は「ホスト機器220は、コンテンツ2314を128バイト単位に分割して再送信せよ」などの要求を含んだ応答を返す。コンテンツ2314のデータを一括して受け取って、その復号処理を実行し、処理結果（成功か失敗か等）を含んだ応答を返してもよい。これにより、ICカードチップ150による復号処理が実行不能になることを防止することができる。ICカードチップ150は、これに加えて、さらに「コントローラチップ120は、復号したコンテンツをフラッシュメモリチップ130にライトせよ」という要求を含んだ応答を返してもよい。そうすれば、ホスト機器220が、コンテンツをフラッシュメモリチップ130にライトするため、MMC110に対してライトコマンドを送信する手間を省くこともできる。

【0099】以下、ICカードチップ150がコントローラチップ120を通じて要求する何らかの処理のことを、「外部処理」と一般化して呼ぶこととする。

【0100】図38を用いてMMC110が外部処理を実行する手順を詳細に説明する。まず、コントローラチップ120はセキュリティ処理のためのICカードコマンドを発行する(3801)。ICカードチップ150はそのコマンドを解析し、外部処理が必要かを判断する(3802)。例えば、前述例のように、ICカードコマンドで入力されたデータの量とICカードチップ150内のワークメモリの空き容量とを比較し、データ量がメモリ容量より大きい又は以上であれば外部処理が必要と判断し、データ量がメモリ容量より小さい場合は外部処理が不要と判断する。手順3802の結果、外部処理が必要ならば、外部処理要求コードを含むレスポンスを送信する(3803)。外部処理要求コードとは、コントローラチップ120に外部処理を要求したいということを告知するための符号である。ICカードチップ150が準拠するISO7816のコマンド規格によれば、I/O端子157からシリアルデータとして出力される、ICカードコマンドのレスポンス信号は、コマンド処理状況を示す「ステータスワード」と呼ばれる例えば2バイトの符号化データをその末尾部（又は前頭部）に含む。これは主に、コマンド処理結果がエラー（6Xh、XXh（Xはエラー内容に依存））であるか、成功（90h、00h）であるかをICカードチップ外部に示すために使用される。外部処理要求コードは、このステータスワードを利用するのが好ましい。すなわち、ステータスワード=91h、YYhを外部処理要求コードと定義する。ここで、YYhは、外部処理の内容を示したデータの長さがYYh（16進数）バイトであるこ

とを予告している。手順3803の次に、コントローラチップ120は、レスポンスを受信し、その受信したレスポンスを分析し、外部処理要求コードが含まれるかを調べる(3804)。外部処理要求コードが含まれているならば、外部処理内容を読み出すICカードコマンドを発行する(3805)。そして、YYhバイトの外部処理内容を含む応答データを待つ。ICカードチップ150がYYhバイトの外部処理内容を含む応答データを送信する(3806)と、コントローラチップ120は受信した外部処理内容を分析する(3807)。そして、コントローラチップ120は、外部処理内容に応じて、フラッシュメモリチップ130またはホスト機器220に対して、上の例で記したような外部処理を指示する(3808)。フラッシュメモリチップ130またはホスト機器220は、指示された外部処理を実行する(3809)。その後、コントローラチップ120は、その外部処理の結果を示すデータを作成し(3810)、ICカードチップ150に結果通知コマンドを発行して外部処理の結果を送信する(3811)。ICカードチップ150は、外部処理の結果を分析し、正しく処理されたか、または、さらなる外部処理が必要かどうかを判断する(3812)。そして、その判断に応じたレスポンスを送信する(3813)。コントローラチップ120は、そのレスポンスを分析する(3814)。そのレスポンス中のステータスワードがエラー（6Xh、XXh）であれば、外部処理はエラーであることを意味する。それが成功（90h、00h）であれば、外部処理は正常に終了したことを意味する。もし、ステータスワードが外部処理要求コード（91h、YYh）であれば、コントローラチップ120は、手順3805に戻って、次に要求される外部処理内容を読み出す手順から再び始める。

【0101】なお、手順3806において送信する外部処理内容の表記法としては、コントローラチップ120とICカードチップ150との間であらかじめ定義された符号化ルールを利用することが好ましい。符号化ルールとしては、例えば、「フラッシュメモリチップ130」を01h、「ICカードチップ150」を02h、「直前に送信されたデータ」を11h、「ライト処理せよ」を22h、「128（80h）バイト単位に分割して送信せよ」を2380hなどとあらかじめ定義しておく。この場合、一例として、外部処理内容データに01h、11h、22h、02h、11h、23h、80hという7バイトの符号を設定することは、「たったいま送信したデータをフラッシュメモリチップ130にライトし、それを128バイト単位に分割してICカードチップ150に送れ」という内容を意味している。このような符号化ルールは、コントローラチップ120内部に格納しておいたものを参照してもよいし、フラッシュメモリチップ130に格納しておき、必要に応じてそこか

ら読み出して参照してもよい。

【0102】また、手順3806において送信する外部処理内容が、フラッシュメモリチップ130に対するリード・ライト処理である場合は、外部処理内容のデータ形式をMultiMediaCard仕様に定義されたリード・ライトコマンドと同じ形式にしてもよい。これは、ホスト機器220がMMC110に対して発行するコマンドと同じ構造のデータ形式となるため、コントローラチップ120内部にあるMMCコマンドの分析手段に外部処理内容のデータを直接入力するだけで外部処理（フラッシュメモリに対するリード・ライト処理）が実行できるため、効率的な処理が実現できる。

【0103】また、手順3806において送信する外部処理内容が、ホスト機器220に対して要求する処理である場合は、外部処理内容のデータ形式をMultiMediaCard仕様に定義されたコマンドのレスポンスと同じ形式にしてもよい。そうすれば、ホスト機器220内部にあるMMCレスポンスの分析手段に外部処理内容のデータを直接入力するだけで外部処理が実行できるため、効率的である。

【0104】ホスト機器220が通信機能を持つような機器であるならば、ホスト機器220に対する外部処理として、「遠隔地のサーバにデータを送信せよ」といった内容にすることも可能である。これにより、ホスト機器220内部にあるサーバ通信プログラムの一部を削減できるため、ホスト機器220内部のプログラムメモリを節約できる。

【0105】ホスト機器220がXML(eXtensible Markup Language: 拡張可能なマーク付け言語)に対応したブラウザ機能を持つような機器であるならば、ホスト機器220に対する外部処理として、「データをブラウザによってディスプレイに表示せよ」といった内容も可能である。このとき、ICカードチップ150が送信する表示データをXML言語で書かれたものにすれば、ホスト機器220内部で表示データ変換をおこなわなくてもよいので、効率的な表示処理が実現できる。

【0106】本発明によれば、ICカードチップが他のチップに対して能動的に処理を要求することにより、ICカードチップによるセキュリティ処理において、フラッシュメモリやホスト機器など外部のデバイスを有効に活用して、より大きなサイズのデータを処理することができるため、利便性の高いセキュリティシステムを実現することができる。

【0107】

【発明の効果】本発明によれば、ICが実行するためのプログラムやデータ、ICを管理するための情報をIC外部の不揮発性メモリに保持するため、IC内の記憶容量（例えば、ROMやEEPROM）が小さい場合にも、ICが多くの処理を実行できるという効果を奏す

る。

【0108】本発明によれば、ICが実行する処理の一部をIC外部のコントローラが実行するため、IC内の記憶容量（例えば、RAM）が小さい場合にも、ICが多くの処理を実行できるという効果を奏する。

【図面の簡単な説明】

【図1】 本発明を適用したMMCの内部構成を示す図である。

【図2】 本発明を適用したMMCのホスト機器の内部構成、およびホスト機器とMMCとの接続状態を示す図である。

【図3】 ICカードチップのコールドリセット時の信号波形を示す図である。

【図4】 ICカードチップのウォームリセット時の信号波形を示す図である。

【図5】 ICカードチップのICカードコマンド処理時の信号波形を示す図である。

【図6】 ICカードチップの非活性化時の信号波形を示す図である。

【図7】 ホスト機器によるMMCへのアクセスを示したフローチャートである。

【図8】 ICカード制御パラメータとそれに対応するICカードへの処理内容を示す表である。

【図9】 ICカードチップに対する第1次ICカード初期化の詳細なフローチャートである。

【図10】 ICカードチップに対する第2次ICカード初期化の詳細なフローチャートである。

【図11】 非活性化状態のICカードチップに対するICカード初期化時の信号波形を示す図である。

【図12】 活性化状態のICカードチップに対するICカード初期化時の信号波形を示す図である。

【図13】 ICカードチップによるセキュリティ処理の詳細なフローチャートである。

【図14】 セキュリティ処理要求ライトコマンドを処理するときの信号波形とフラッシュメモリチップアクセスを示す図である。

【図15】 ICカードチップによるセキュリティ処理実行時の信号波形とフラッシュメモリチップアクセスの一例を示す図である。

【図16】 セキュリティ処理結果リードコマンドを処理するときの信号波形とフラッシュメモリチップアクセスを示す図である。

【図17】 インタフェース直通モードにおけるMMC外部端子とICカードチップ外部端子の対応関係を示す図である。

【図18】 インタフェース直通モードへ移行する処理とインタフェース直通モードから復帰する処理のフローチャートである。

【図19】 インタフェース直通モードへ移行する処理時の信号波形を示す図である。

【図20】 インタフェース直通モードから復帰する処理時の信号波形を示す図である。

【図21】 フラッシュメモリチップの内部構成を示す図である。

【図22】 本発明を適用したMMCの内部構成を簡単に示す図である。

【図23】 本発明を適用したMMCをコンテンツ配信に応用した例を示す図である。

【図24】 本発明を適用したSDカードの内部構成を簡単に示す図である。

【図25】 本発明を適用したメモリースティックの内部構成を簡単に示す図である。

【図26】 本発明のICカードチップの内部構成を示す図である。

【図27】 本発明を適用したMMCの内部構成を示し、特にICカードチップの機能的構成の詳細を含む図である。

【図28】 本発明を適用したMMCを銀行取引サービスに応用した場合の、システム構成を示す図である。

【図29】 本発明を適用したMMCを銀行取引サービスに応用した場合の、取引実行時のフローチャートの前半である。

【図30】 本発明を適用したMMCを銀行取引サービスに応用した場合の、取引実行時のフローチャートの後半である。

【図31】 フラッシュメモリチップの内部構成を示

し、特にICカード環境設定情報領域の詳細を含む図である。

【図32】 コマンド処理モジュール群の一例を示す図である。

【図33】 ICカードチップの中のROM、RAM、EEPROMの内部構成を示す図である。

【図34】 コマンド処理モジュールのロード手順およびセキュリティコマンドの実行手順のフローチャートである。

【図35】 モジュール管理情報の詳細および具体例を示す図である。

【図36】 コマンド処理モジュールの更新手順のフローチャートである。

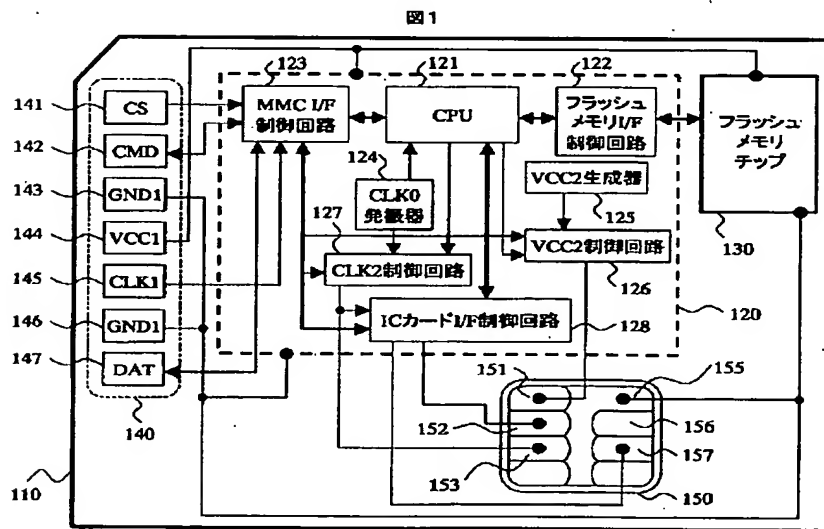
【図37】 コマンド処理モジュールのロードや更新に使用する鍵の説明を示す図である。

【図38】 ICカードチップの要求に従い、コントローラチップがフラッシュメモリチップやホスト機器に処理を実行させる手順を示すフローチャートである。

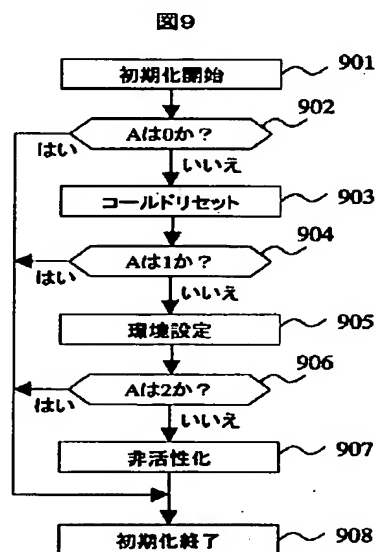
【符号の説明】

110…MMC、120…コントローラチップ、140…MMC外部端子、150…ICカードチップ、151…VCC2端子、152…RST端子、153…CLK2端子、155…GND2端子、156…VPP端子、157…I/O端子、220…ホスト機器、1405…ライトコマンド発行、1906…モード移行時刻、2003…モード復帰時刻。

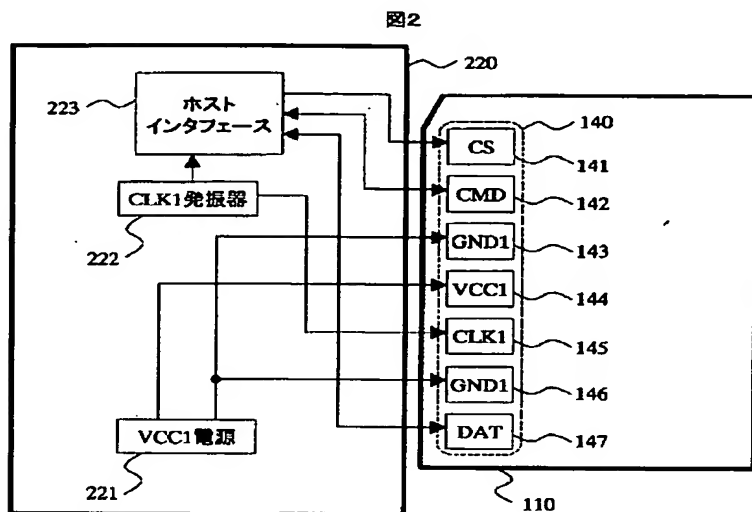
【図1】



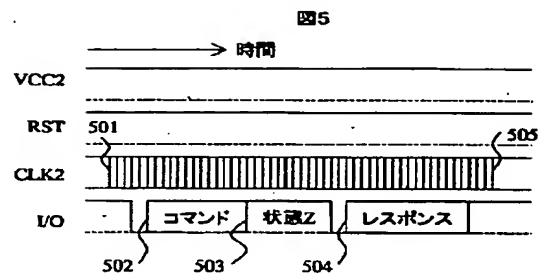
【図9】



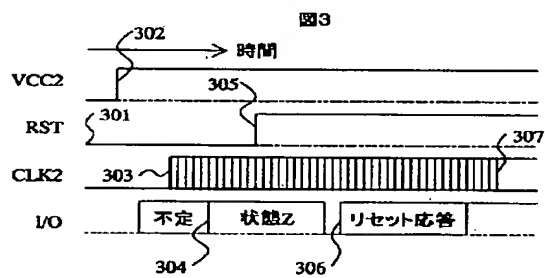
【図2】



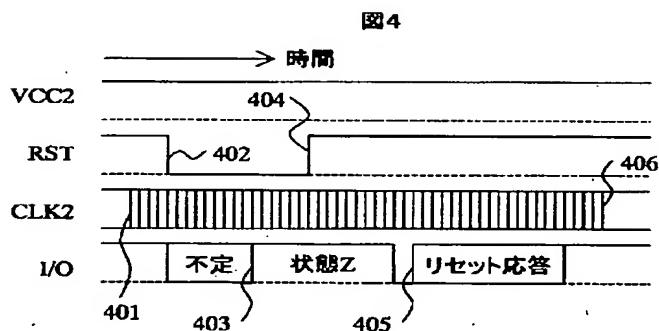
【図5】



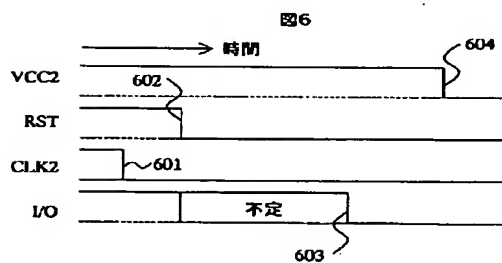
【図3】



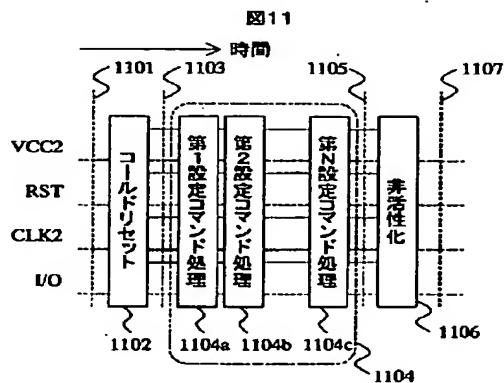
【図4】



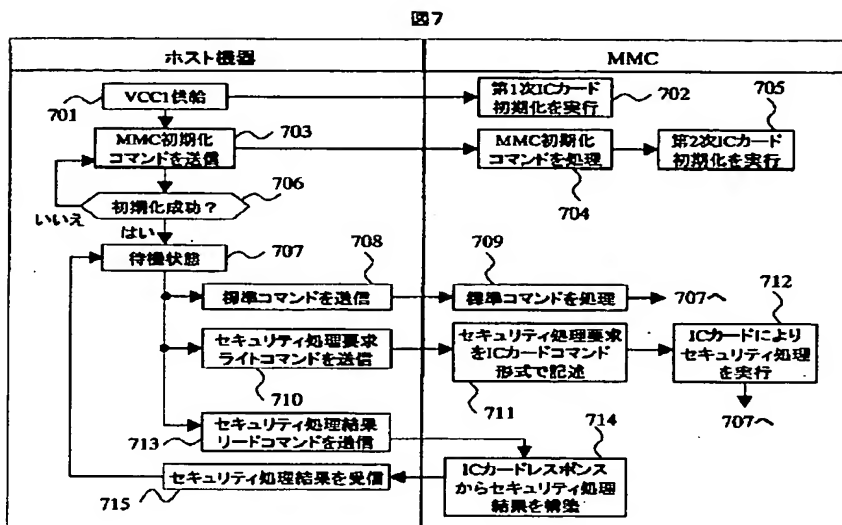
【図6】



【図11】



【図7】



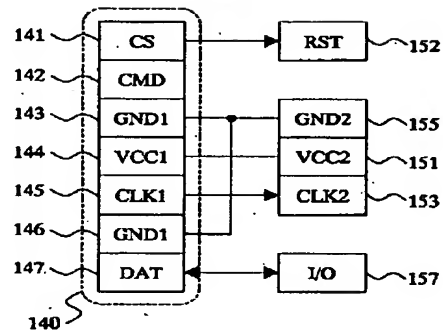
【図8】

図8

ICカード制御 パラメータ	ICカードに対する処理
A=0	MMCのパワーオン時に、何もしない
A=1	MMCのパワーオン時に、リセット
A=2	MMCのパワーオン時に、リセットと環境設定
A=3	MMCのパワーオン時に、リセットと環境設定し、非活性化
B=0	MMCの初期化時に、何もしない
B=1	C=1 MMCの初期化時に、リセット
	C=2 MMCの初期化時に、リセットと環境設定
	C=3 MMCの初期化時に、リセットと環境設定し、非活性化
B=2	C=2 MMCの初期化時に、環境設定
	C=3 MMCの初期化時に、環境設定し、非活性化
B=3	MMCの初期化時に、活性状態ならば、非活性化
D=0	セキュリティ処理後に、非活性化しない
D=1	セキュリティ処理後に、非活性化する

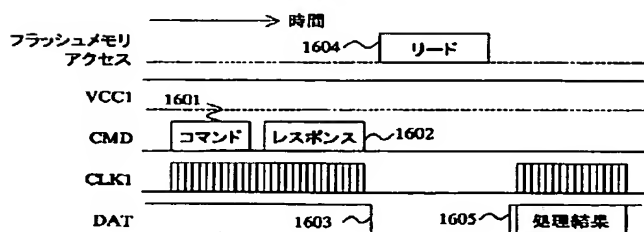
【図17】

図17

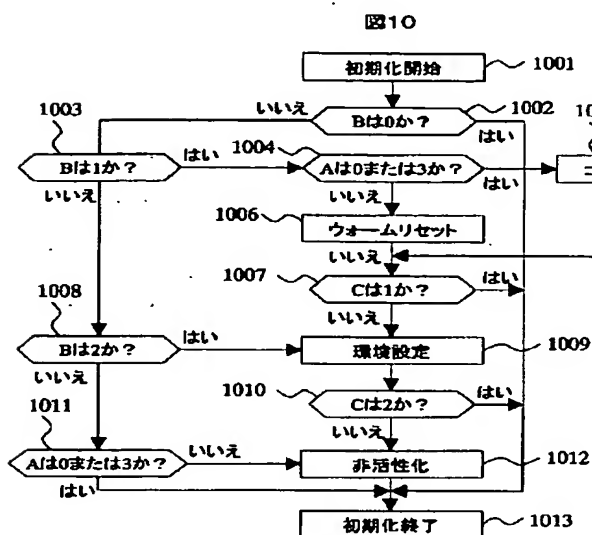


【図16】

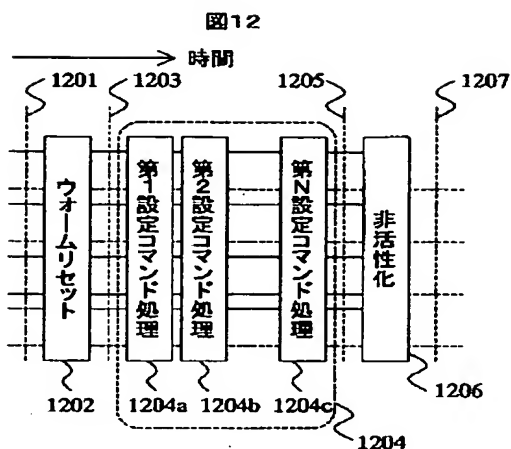
図16



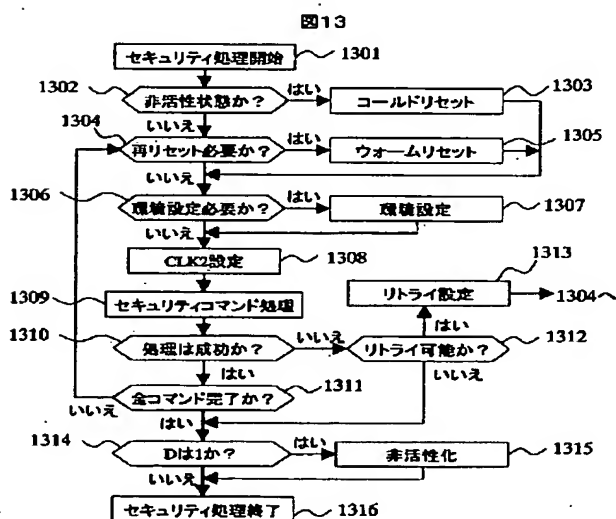
【図10】



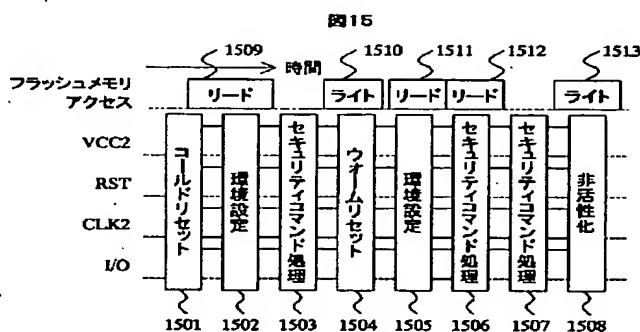
【図12】



【図13】

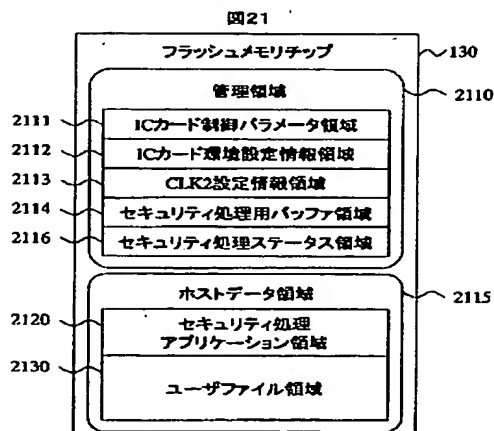


【図15】

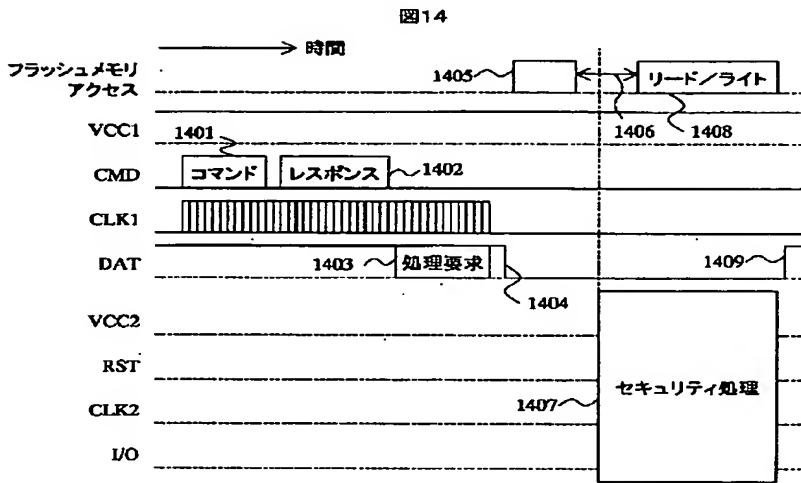


35

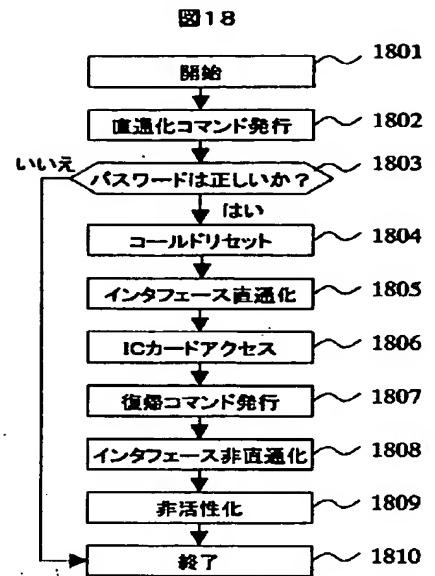
【図21】



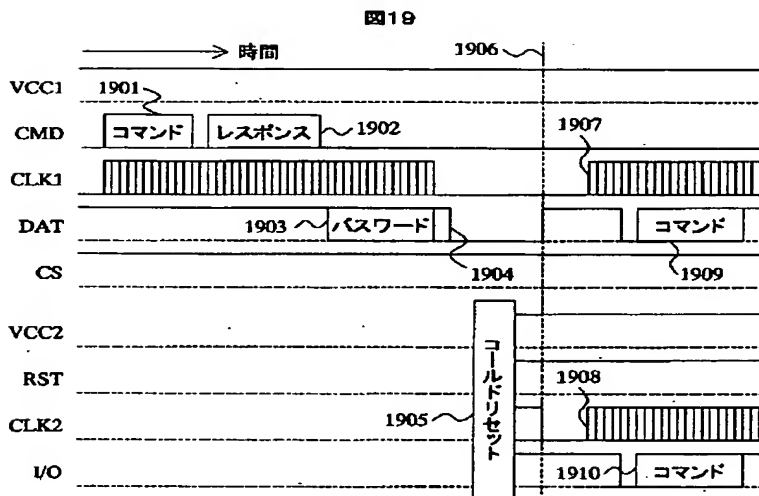
【図14】



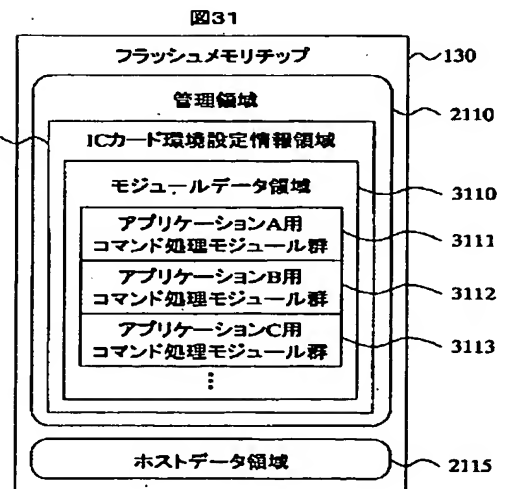
【図18】



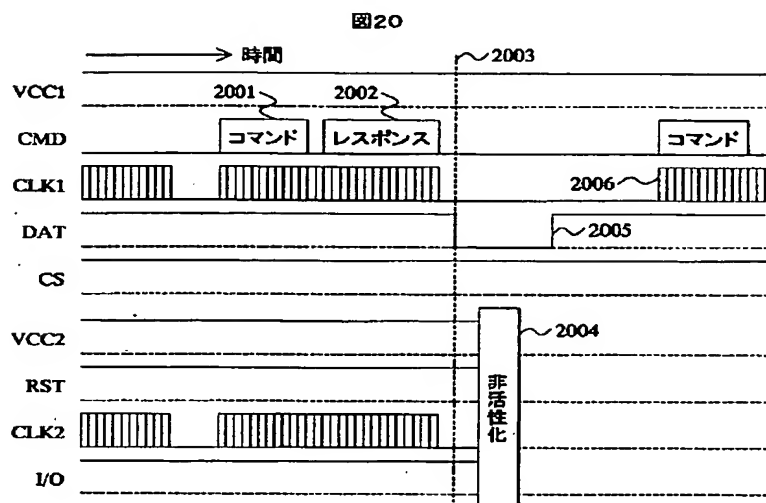
【図19】



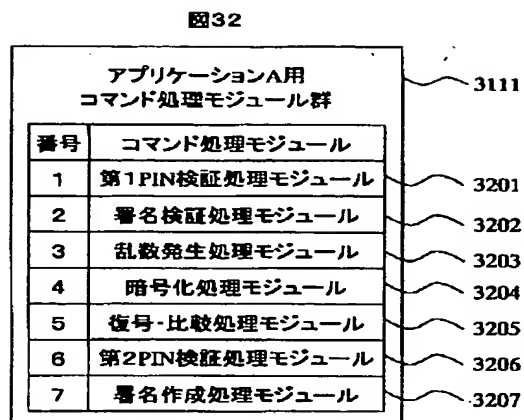
【図31】



【図20】

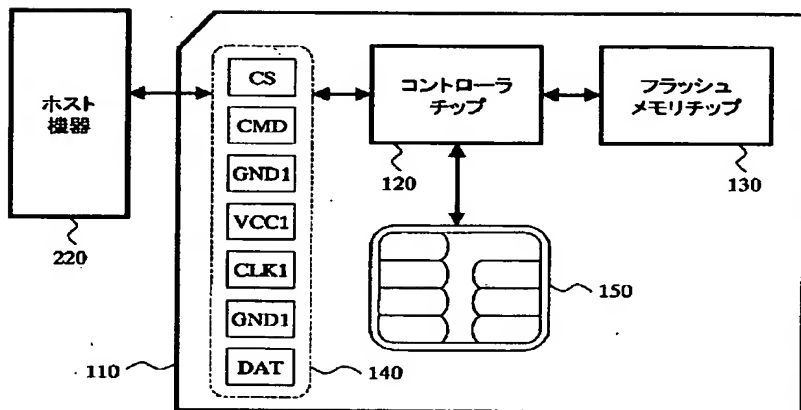


【図32】



【図22】

図22



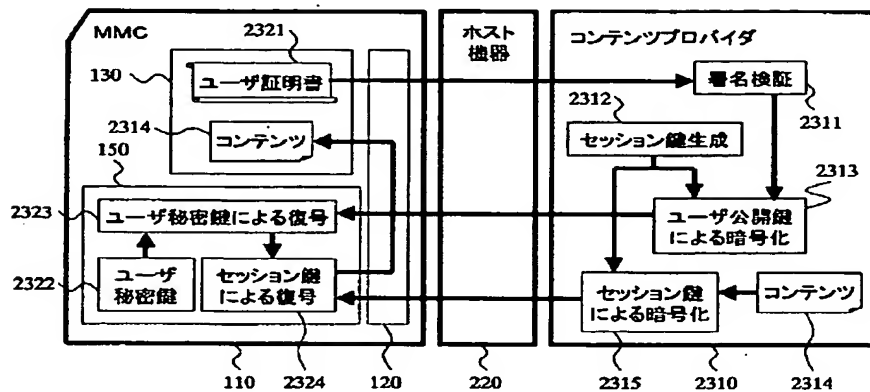
【図37】

図37

鍵の名前	説明
3701 モジュール鍵	コマンド処理モジュールにかけられた暗号を解き、実行可能な形式へ復元するための対称鍵。
3702 アプリケーション秘密鍵	アプリケーション発行者が厳重に管理すべき秘密鍵。コマンド処理モジュール更新時に使用。
3703 アプリケーション公開鍵	アプリケーション秘密鍵に対応する公開鍵。ICカードチップ内の各アプリケーションが保持。コマンド処理モジュールロード時、およびコマンド処理モジュール更新時に使用。

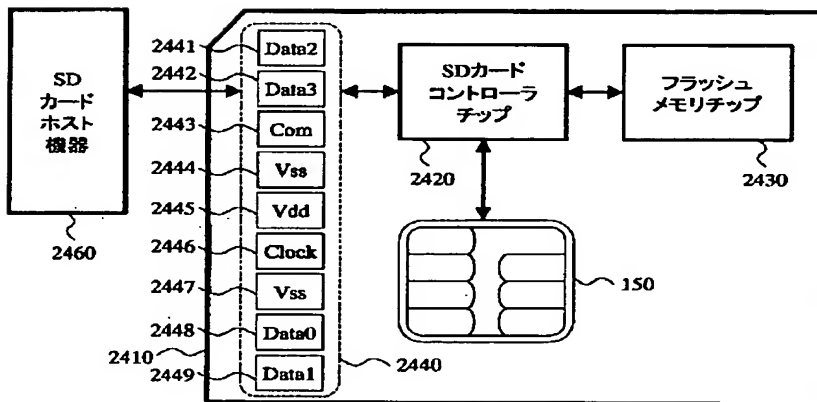
【図23】

図23



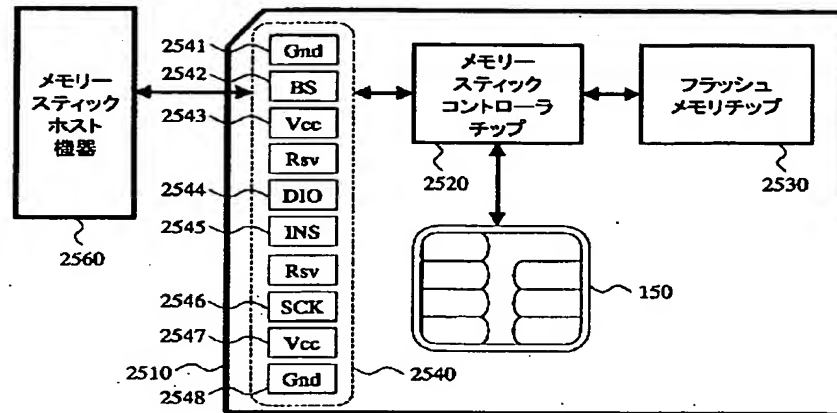
【図24】

図24



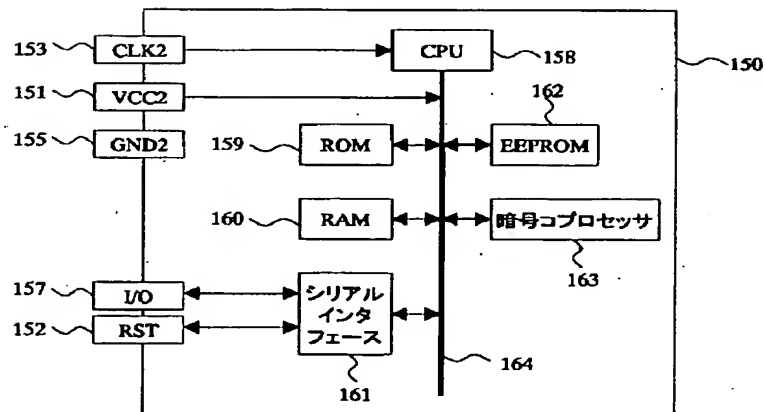
【図25】

図25

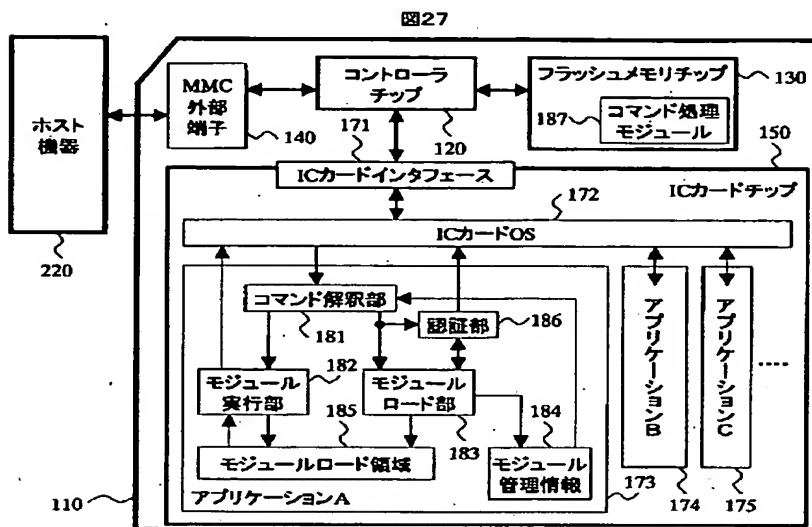


【図26】

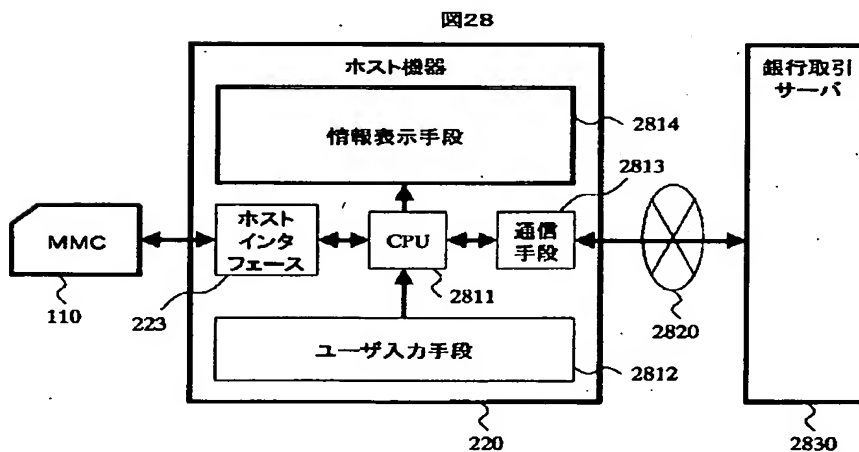
図26



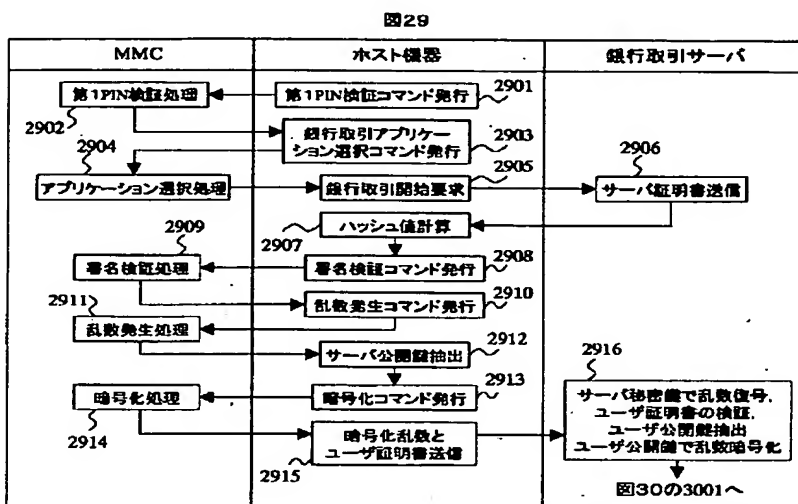
【図27】



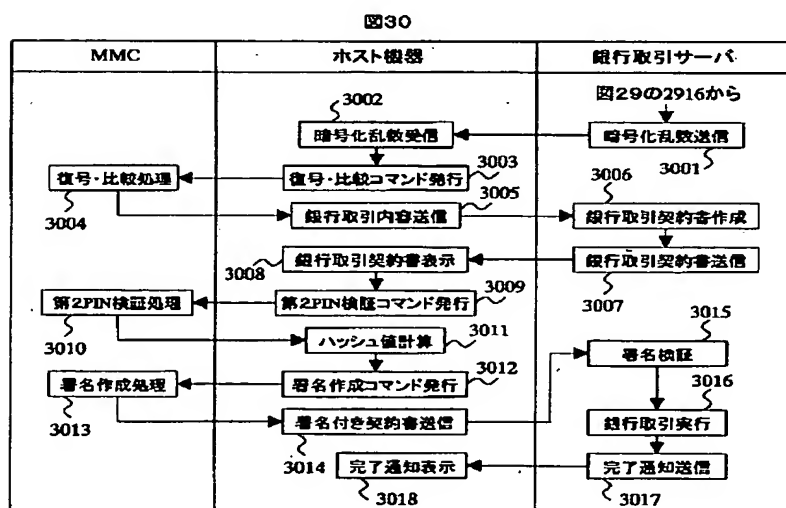
【図28】



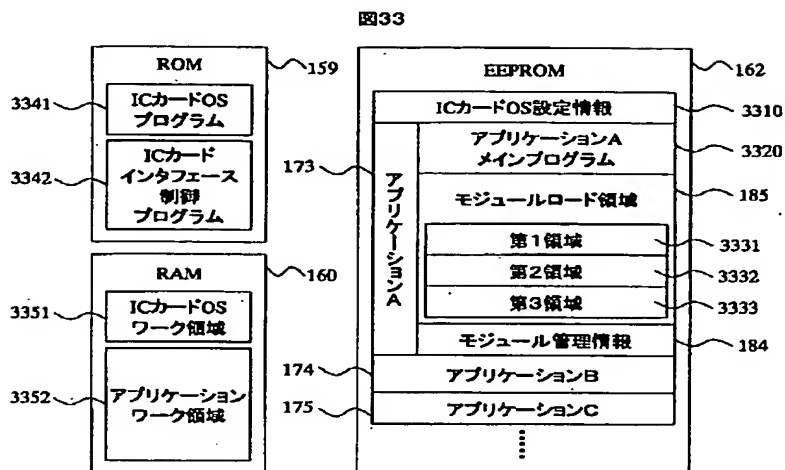
【図29】



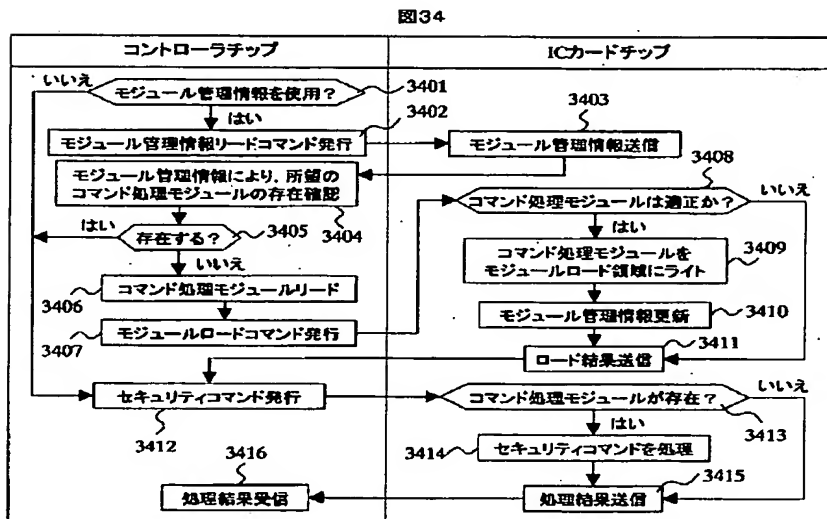
【図30】



【図33】



【図34】



【図35】

図35

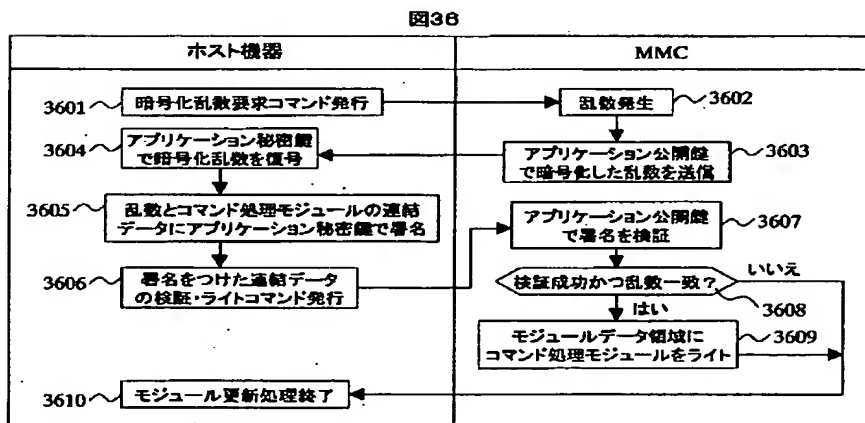
3511 3512 3513 3514 3515 3516 3517

3510 ロード管理 情報	番号	処理内容	サイズ	状態	使用版	最新版	ロード条件
	1	第1PIN検証	280	第2領域	3. 1	3. 1	最新版のみ
	2	署名検証	188	第3領域	1. 5	2. 2	無条件
	3	乱数発生	96	なし		1. 0	無条件
	4	暗号化	256	なし		1. 2	最新版のみ
	5	復号・比較	388	第1領域	1. 4	1. 6	1. 0以上
	6	第2PIN検証	280	なし		3. 1	最新版のみ
	7	署名作成	176	なし		1. 8	1. 5以上

3522

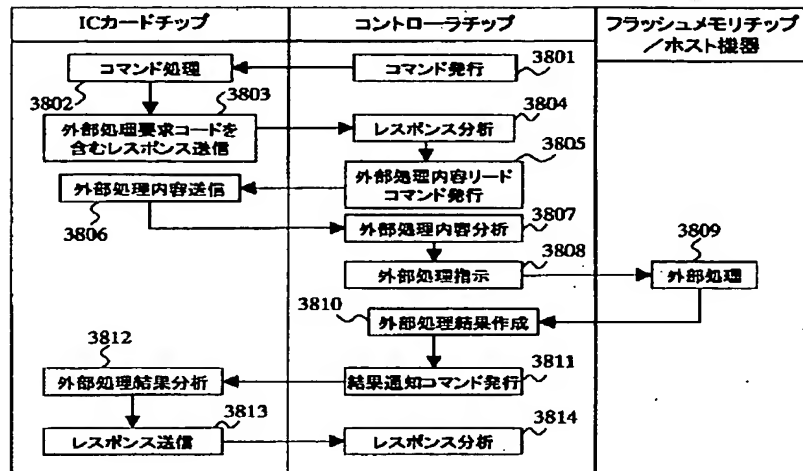
3520 ロード 領域 情報	番号	領域名	サイズ	3521
	1	第1領域	400	
	2	第2領域	320	
	3	第3領域	200	

【図36】



【図38】

図38



フロントページの続き

(72) 発明者 幡野 富久
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 片山 国弘
東京都小平市上水本町五丁目20番1号 株式会社日立製作所半導体グループ内

(72) 発明者 田中 紀夫
神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所金融システム事業部内

25 (72) 発明者 常広 隆司
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

(72) 発明者 木村 光一
神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

30 Fターム(参考) 2C005 MA04 MB01 MB08 MB10 NA02
NA40 NB01 NB04 SA02 SA03
SA11 SA22
5B035 AA06 AA13 BB09-BC00 CA11

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.